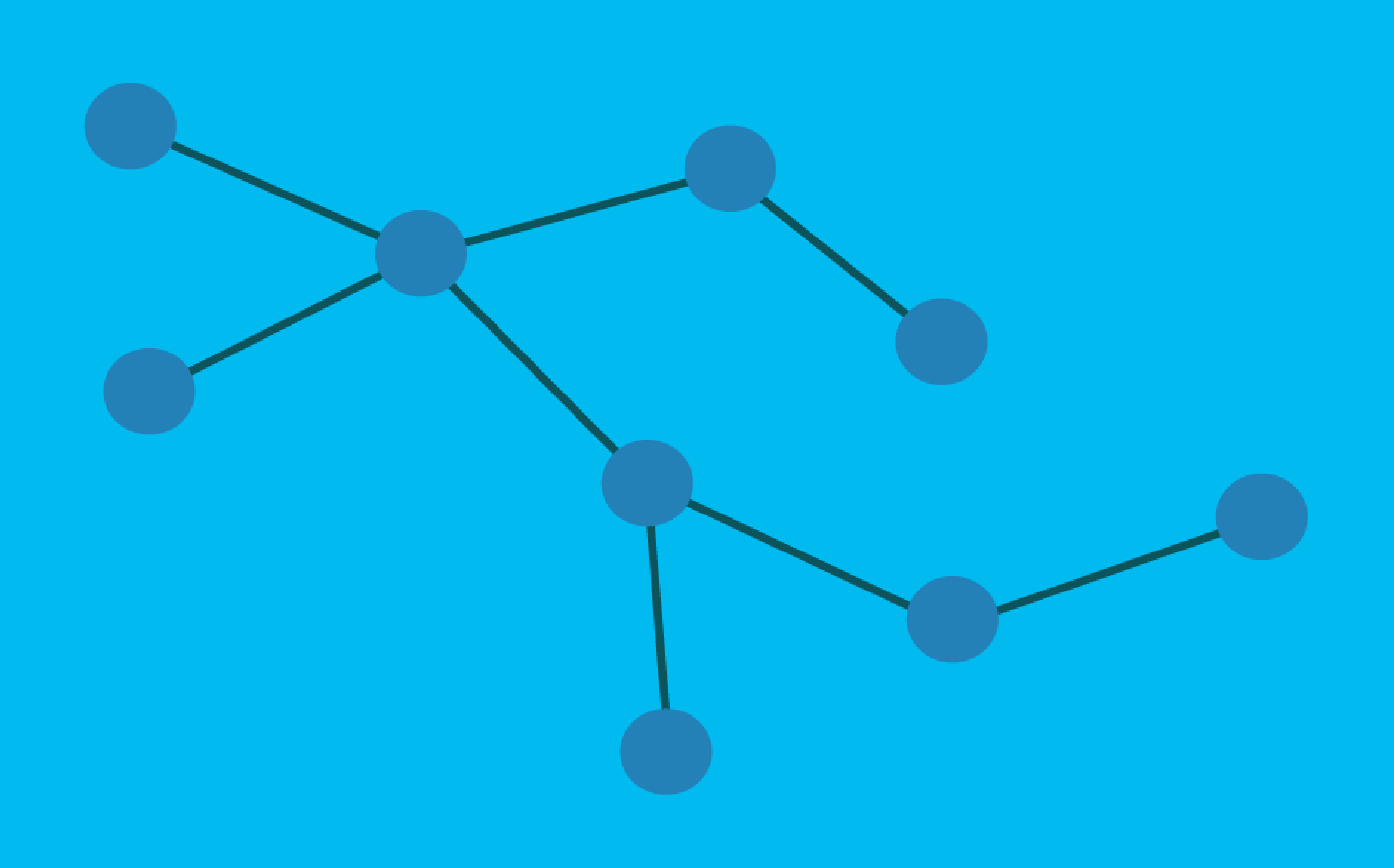
高等院校计算机任务驱动教改教材

数据安全与灾备管理

贾如春　周晓花　主　编

陈新华　王宏旭　吴　粟　副主编

赵克林　康　乐　主　审



清华大学出版社

高等院校计算机任务驱动教改教材

数据安全与灾备管理

贾如春　周晓花　主编
陈新华　王宏旭　吴　粟　副主编
赵克林　康　乐　主审

清华大学出版社
北　京

内 容 简 介

本书基于项目化教学方式编写而成，并引入实际企业案例。主要内容包括：数据存储基础、存储应用环境、数据存储技术、RAID 技术与应用、虚拟磁带库技术、数据灾备与恢复技术、虚拟化技术、灾备系统设计与典型案例分析、数据中心安全运维、云计算应用与云灾备、大数据存储。全书以社会调查、企业岗位需求为基准，以一家真实企业公司在不同发展时期的 IT 架构建设、数据安全与灾备为背景，从数据灾备系统建设到数据中心、数据安全维护进行了详细讲解，并配合详细的案例分析与实训内容，本着“学生能学、教师能教、企业能用”的原则，注重理论知识与实践技能的掌握，不断培养学习者的自信心和自学能力，力求学习者通过学习对此技术领域形成一个比较全面的知识体系，并且掌握数据安全与灾备相关的主流技术和产品管理，从而提高自身技能与分析解决问题的能力，将对知识的理解和实践应用有机地结合为一体，更好地应用于社会服务中。

本书既适合作为计算机相关专业本科、专科学生的教材，也可以作为相关工程技术人员的学习用书。

图书在版编目(CIP)数据

数据安全与灾备管理/贾如春，周晓花主编. --北京：清华大学出版社，2016
高等院校计算机任务驱动教改教材
ISBN 978-7-302-44507-4

Ⅰ. ①数…　Ⅱ. ①贾…②周…　Ⅲ. ①数据处理—安全技术—高等学校—教材　Ⅳ. ①TP274

中国版本图书馆 CIP 数据核字(2016)第 171848 号

责任编辑：张龙卿
封面设计：徐日强
责任校对：李　梅
责任印制：李红英

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62770175-4278
印 装 者：北京国马印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：23.25　　**字　　数**：562 千字
版　　次：2016 年 8 月第 1 版　　**印　　次**：2016 年 8 月第1 次印刷
印　　数：1～2000
定　　价：49.00 元

产品编号：070241-01

前　言

随着“互联网＋”、大数据时代的到来，人类已经从信息技术(Information Technology，IT)时代逐步走向数据技术(Data Technology，DT)时代。业务应用数据的日益更新越来越深刻地影响着企业的经营管理模式，这当中影响最大的就是企业的信息管理模式。一方面极大地促进了企业的发展，另一方面又对海量信息数据的存储和管理提出了新的挑战。对于数据量很大的企业级数据来说，性能好坏非常重要，直接影响到业务的质量。因此如何有效使用数据已成为IT系统建设中极其重要的基础部分，并已成为计算机领域里相对独立的分支学科。

大数据时代的到来，也对数据安全提出了更高的要求，对于信息化应用而言，灾备系统的建设已成为热点。随着2007年《信息系统灾难恢复规范》(GB/T 20988—2007)正式成为国家标准，许多用户对数据灾备从观望、徘徊转向实际应用。随着虚拟化的广泛应用、云计算的出现，以及数据量每年的高速增长，数据类型和来源的多样化，使得在如此复杂的环境下如何高效、安全、可靠、完整地使用和保存宝贵的数据资料已成为当务之急，只有使用高性能计算机网络存储系统，才能从根本上解决企业日益增长的需要。而自动化的存储管理手段，不仅可以解决现有企业关键数据的存储和管理的需求，而且可以同时为网络上各种工作站的用户提供数据的备份，减轻系统管理员的负担，有效地保护宝贵的数据及人力资源，这样在不幸遇到灾难后，就可以很迅速地恢复数据，使整个系统在最短的时间内重新投入正常运行。

本书响应国家安全号召，坚持“自主开发、安全可控”的原则，基于国内众多知名企业如精容数安、人大金仓、聚比特科技等具有多年计算机系统建设、存储系统管理、数据安全管理、数据备份系统建设等真实项目案例累积的经验基础上所提供的资料，致力于为用户提供最完整的解决方案，不仅能够支持企业自身的操作系统平台，而且能够支持COMPAQ、IBM、Microsoft、SGI、SUN等各种主流操作系统平台和多种数据库和应用平台，涵盖从高可用性、远程/异地容灾、数据整合到网络数据备份、零停机备份以及长期归档等数据管理和数据保护的解决方案。

本书内容概况如下。

(1) 数据存储概述篇。本篇主要针对计算机系统中与存储相关的技术进行介绍，包括存储器的组成结构、工作原理、存储管理系统以及不同存储

产品技术路线等内容。

（2）数据灾备概述篇。本篇主要是针对数据灾备技术的不同方案类型以及在虚拟环境下如何实现灾备技术部署等内容进行介绍。

（3）数据灾备应用篇。本篇主要针对整个行业典型的灾备技术解决方案进行介绍和模拟，以及对大型数据中心运维中有关数据安全的运维服务进行介绍。

（4）数据灾备趋势篇。本篇主要讲述云计算的相关技术、设施框架类型以及各种云环境下的云灾备和数据迁移服务技术。

本书不仅适合作为本科院校及职业院校计算机类、通信类相关专业的教材使用，也可作为从事系统集成、数据容灾备份等领域工作的相关技术人员的参考用书。

本书由贾如春负责总体策划设计及统稿，并与企业专家周晓花共同担任本书主编，陈新华、王宏旭、吴粟担任本书的副编辑，全书由赵克林教授及企业专家康乐主审。感谢精容数安科技有限公司给予作者的支持与帮助，同时感谢其他高等院校中各位教师对本书提出的修改建议。

由于作者水平有限，本书涉及的知识面较广，虽然编者已经尽了最大努力，但书中难免存在错漏之处，欢迎广大读者批评指正。

编　者

2016 年 6 月

目　录

第一篇　数据存储概述

第二篇　数据灾备概述

第四篇　数据灾备趋势



第一篇

数据存储概述



第 1 章　数据存储基础



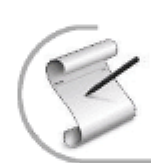
任务目标

- 了解数据存储的概念与发展历程；
- 了解数据存储的原理与物理结构；
- 了解数据存储的基本介质与技术；
- 了解数据灾难与容灾基础。



项目背景

存储技术作为信息技术的核心之一，一直伴随并推动着 IT 业务各方面技术的协同发展，是当今 IT 领域中少数发展最为迅速的热点之一。纸的发明记载了人类的历史和文明，现代信息存储技术则大大超越了纸张记录的含义。如果说信息代表的是生存和生命、进取和发展，那么信息的价值是无可估量的，而存储作为信息的载体使信息的价值得到实现和增值，也就是说，存储的数据才是现代人类社会的真实财富所在。



项目描述

聚比特科技有限公司自成立以来一直从事互联网和电子商务等业务，但是随着信息化的发展，出现了很多瓶颈，越来越影响着企业的经营管理模式，这当中影响最大的便是企业的信息管理模式，随着应用互联网和电子商务业务应用的增长，企业信息数据呈爆炸性增长，一方面极大地促进了企业的发展，另一方面又对海量信息数据的存储和管理提出了新的挑战。如何有效地解决这些问题，成为企业及研究人员正在思考的问题。数据存储直接影响着行业的发展，也影响着企业自身的竞争力。



项目分析

如何高效、安全、可靠、完整地使用和保存宝贵的数据资料，又如何从这些浩瀚如海的信息中顺利地找到所需要的信息，成为聚比特科技有限公司的当务之急，只有使用高性能计算机网络存储系统，才能从根本上解决企业数据日益增长的需求。



项目实现

通过企业信息化专家分析形成一种自动化的存储管理手段，不仅可以解决现有企业关键数据的存储和管理需求，而且可以同时为网络上各种工作站用户提供数据备份的解决方案，减轻系统管理员的负担，有效地保护宝贵的数据及人力资源。并且当数据遇到灾难后，

可以在第一时间内迅速地恢复数据,使整个系统在最短的时间内重新投入正常运行。

1.1 数据存储发展历史

存储为信息记录,是伴随人类活动出现的技术。自世界上第一台计算机问世以来,计算机的存储器件也在不断地发展更新,从一开始的水银延迟线存储器、磁带、磁鼓、磁芯,到现在的半导体存储器、磁盘、光盘、纳米存储等,无不体现着科学技术的快速发展。伴随着计算机的飞速发展,计算机上面的存储器也出现了翻天覆地的变化。也可以说,存储技术的飞跃,也促进了计算机的发展。

1.1.1 信息数据发展

对于IT行业,存在一个耳熟能详的定律,即摩尔定律。摩尔定律定义如下:从现在开始的每18个月,微处理器的性能提高一倍,而价格下降一半。而在信息世界中,根据各种应用的普及以及大量数据的产生,定义于硬件的摩尔定律对于数据的增长量的预测也同样有用。在当今的信息环境下,数据将随着时间的推移而呈几何级数增长,庞大的信息使得人们在信息存储方面所花费的管理和维护开销大大增加,来自于IDC数字宇宙研究中心预测过全球的数据量,如图1-1所示。不仅如此,如何安全、合理地保存这些新增的数据,又如何从这些浩瀚如海的信息中顺利地找到人们所需要的信息,这些问题成为摆在数据管理人员面前的难题。

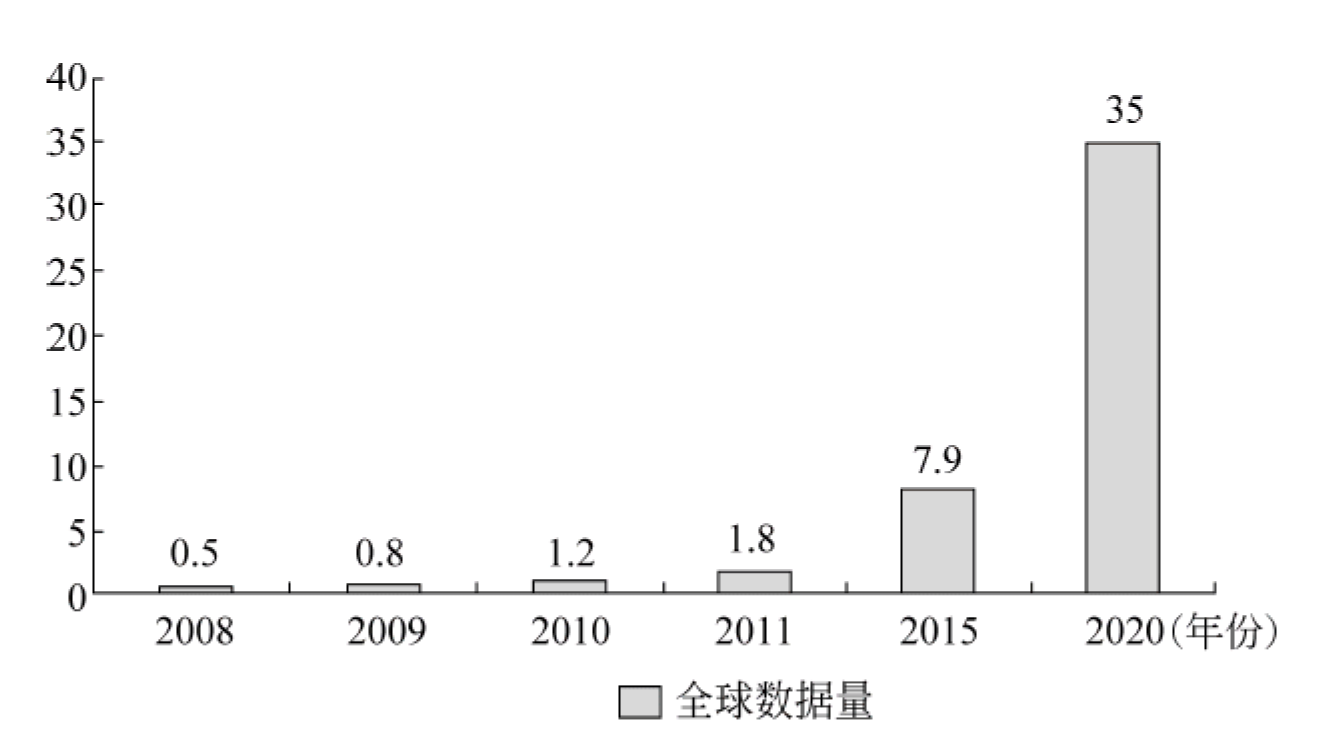


图1-1 全球数据量急剧增长

自计算机产生以来,数据信息的处理能力大幅提高,这也使得数据信息爆炸式增长。而在1973年5月22日以太网络发明之后,更多的数据信息通过网络传递于分布在全球各地的信息系统当中。当大部分的IT系统管理人员一直把注意力放在如何提升主机、网络的数据处理性能的时候,部分研究人员却已经将注意力放在了数据的存储上。由于种种历史原因,数据被分割成杂乱且分散存放的“数据孤岛”,需要的信息无法得到充分的利用,并且设备的充分利用和资源的共享也极为困难。在这样的背景下出现的存储系统,从本质上解决了数据集中存储、共享和管理以及分布备份的问题,为整个系统的可靠、便捷应用提供了坚实的基础。

越来越多的 IT 组织将存储系统的规定与建设视为其现有以及待建的应用系统赢得差异化竞争优势的战略性重要因素。越来越多的 IT 组织都已经认识到了数据在企业中所起到的关键性作用，进而清晰地认识到，企业需要强大、先进、便捷的存储基础设施来支持企业的数据管理。正是基于此，在 IT 设备采购过程中，更多的采购人员会将服务器和存储设备分别进行采购，这样有助于确保存储解决方案不再被单纯地视为新建应用系统的附属品。对于现代企业而言，这是一种新的尝试的开始，并且这种尝试可能会使得企业在应用系统建设的初期就确立其优越性和先进性。而随着存储市场上技术、产品的日益成熟，越来越多的 IT 专业人员将深入地认识存储的价值，主动去掌握相关的技能，并努力地将这一潮流继续推进。

伴随着数字化的发展，个人对数据空间的需求逐渐增加，包括普通的照片、DV、流行音乐、经典电影、重要数据的保存等。目前，国内大约有 6800 万的宽带账号，而且这个数字的增长会越来越快，这些戈恩用户需要更大、更有效的设备来存储、管理、保护和分享他们大量的数据信息。

现在企业对于数据的依赖性愈加严重，现代的企业中存在大量的 IT 应用系统用于处理各种各样的数据——产生数据、销售数据、人事信息、客户信息等。大量应用系统的上线直接导致海量数据的产生，而美国“9·11 事件”表明，这些数据甚至直接关系着企业的生死存亡。所以，如何有效、安全地存储这些海量数据，已经成为当下企业面临的头等大事。

1.1.2　存储的基本概念

所谓存储，是指将数据信息整合并保存在某种介质上的一个过程的结果。对于传统的计算机系统而言，存储并不是一个孤立的系统，是依附于传统应用的一个系统组件。而随着计算机系统的发展和数据的爆炸式增长，促使存储系统的发展朝着独立化的方向进行。

(1) 存储设备是用于存放数据信息的设备和介质。

(2) 存储是一个系统，等同于计算机系统中的外部存储系统。

(3) 存储系统独立化是计算机技术发展的必然结果。

1.2　数据存储原理

1.2.1　存储的基本原理

存储就是根据不同的应用环境通过采集合理、安全、有效的方式将数据保存到某些介质上并能保证有效的访问，从而向用户提供一套数据存放和读取的解决方案。

存储的特征如下：

(1) 它是数据临时或长期驻留的物理媒介；

(2) 它是保证数据完整安全存放的方式或者行为。

传统计算机存储系统的组成如图 1-2 所示。

在计算机系统中，存储分为外部存储和内部存储。在传统的计算机存储系统中，存储通常是由计算机内置的硬盘完成，而采用这样的设计方式，硬盘本身的缺陷很容易成为系统的性能瓶颈，并且由于机箱内有限的空间，限制了硬盘数量的扩展，并且同时也对机箱内的散

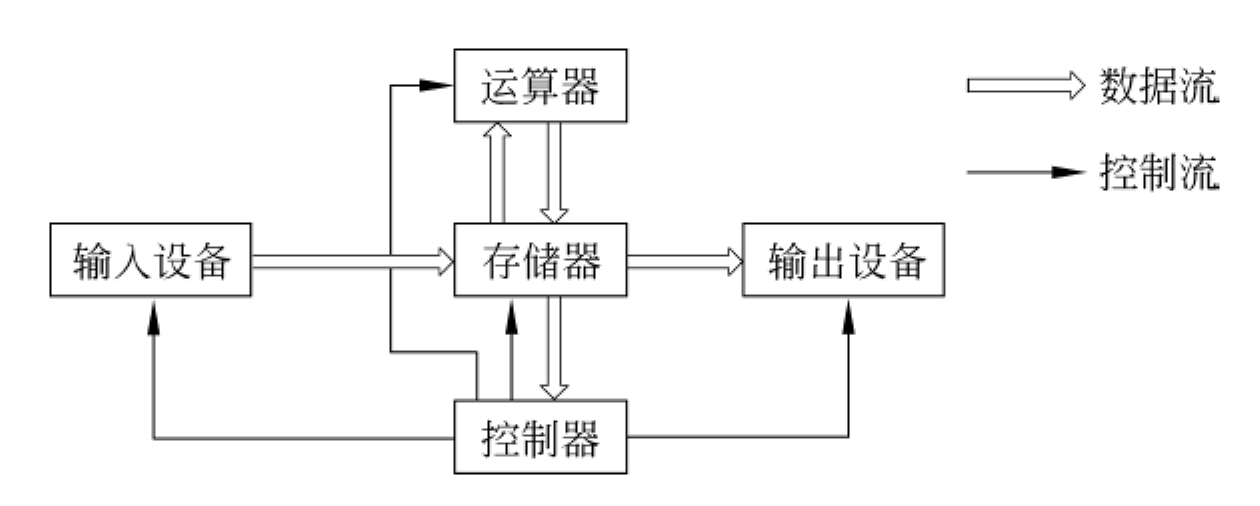


图 1-2 传统计算机存储系统的组成

热、供电等提出了严峻的挑战。再加上不同的计算机相互独立,各自使用内置的硬盘,导致从总体看来存储空间的利用率较低,并且分散的数据也不利于数据的共享和备份工作。

在传统的 C/S 架构中,无论使用的是何种协议,存储设备都直接与服务器相连接,在这样的结构下,对存储设备上保存的所有数据的任何读写操作,都必须由服务器来进行,这样的处理方式给服务器带来沉重的负担。外部存储系统的出现,彻底将服务器从烦琐的 I/O 操作中解放出来,使服务器更加专业化,使之仅仅承担应用数据的操作任务,以便充分地释放自身的潜能。

1.2.2 常见存储设备及其组件

在计算机系统中,存储网络建立了独立的基于网络的存储架构,增加了现有 C/S 的计算机拓扑架构,从而从整体上改变了传统计算机存储系统的模型。存储网络允许存储设备直接连接到现有网络上,也可以通过专门的存储网络进行连接。这一技术给传统的存储配置方案带来了两个重要的变化。

(1) 存储网络与存储设备、服务器以及客户机之间建立了更多的直接访问路径,使用户能够绕过大量的服务器 I/O 操作而直接与数据发生联系,从而避免了对服务器进行不必要的访问。

(2) 存储网络使商务应用系统能够以更高的效率访问数据。换言之,存储网络使得应用系统能够更有效地共享数据,并赋予服务器更为强大的数据连接能力。

1. 常见存储组件

当今的存储技术不是一种单独的技术,实际上,完整的存储系统是由一系列组件构成的。

目前,存储系统主要分为硬件架构、软件组件以及实际应用时的存储解决方案三部分。而硬件架构部分又包括外置的存储系统,即存储设备,比如磁盘阵列、磁带库等。除此以外,存储连接设备用于互联存储设备和主机系统,以及用于对整个存储系统做管理用的存储管理设备,比如对硬盘框做管理用的控制框等。

因为软件组件的存在,使存储设备的可用性得到了大大的提高,从创建不同的 RAID 级别的存储资源到数据的镜像、复制,自动的数据备份等,数据操作都可以通过对存储软件的操作来完成。

一个设计良好的存储解决方案,是人们进行数据存储工作更加简单易行的最佳保障,设计优秀的存储解决方案,不仅可以使存储系统实际部署的时候更简单容易,也更能降低客户的总体拥有成本(TCO),使用户的投资能得到良好的保护。如图 1-3 所示为一个存储解决方案。

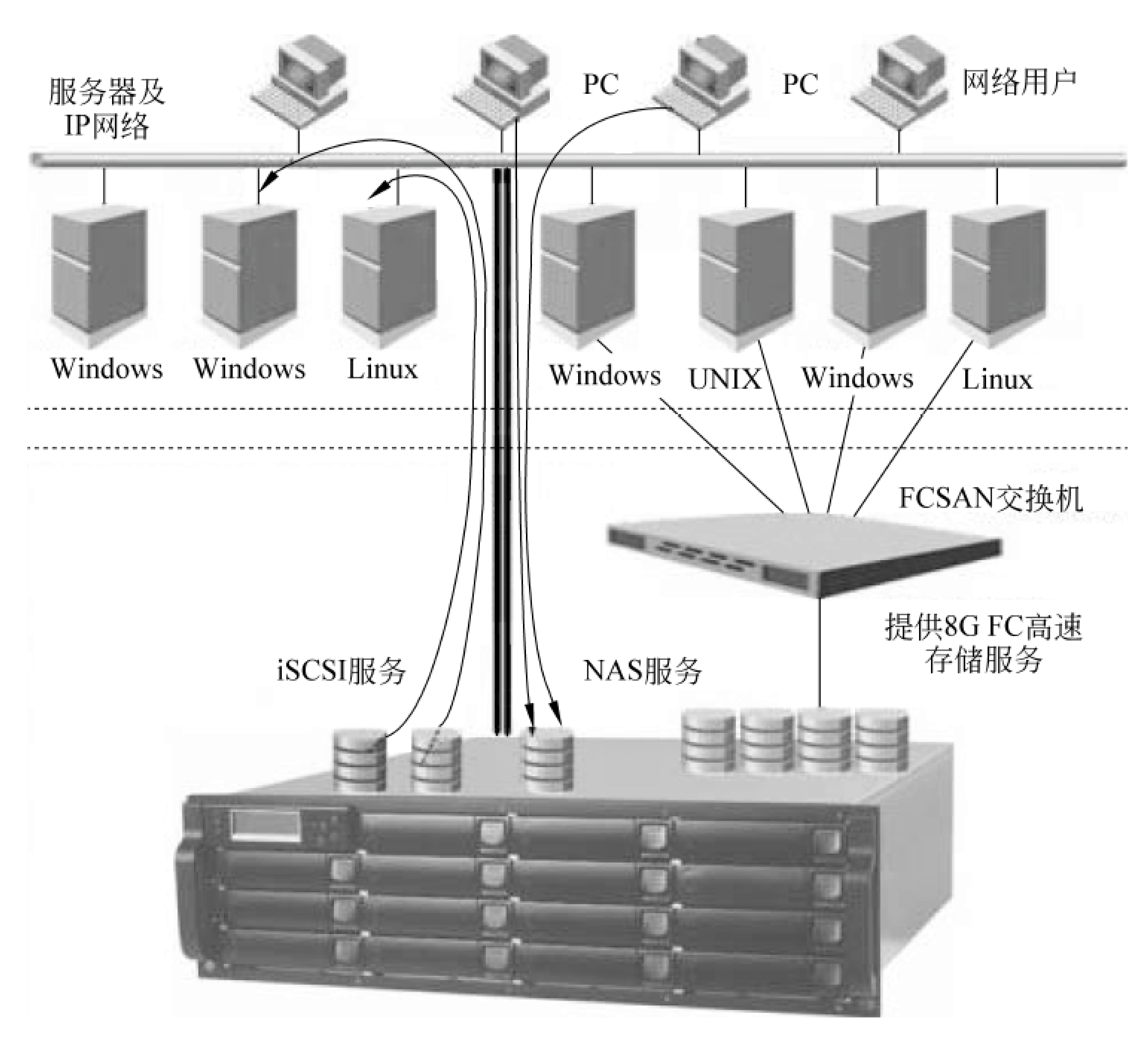


图 1-3　存储解决方案

2. 常见存储介质

常见的存储介质如图 1-4 所示。



图 1-4　常见的存储介质

(1) 硬盘：硬盘是一种非易失性的、可随机编址、可重写的，且使用磁性介质盘片作为存储介质的数据存储设备，其特点如下。

① 寻址访问、数据存储速度快，但成本高。

② 适合做快速响应访问的场合。

(2) 磁带：磁带是一种按数据发送的顺序将数据写入，并且能够以数据的存储位置顺

序将数据读出，读写速度快，容量大，脱机存放容易，成本低。

① 顺序读写、读写速度快、容量大、脱机存放容易、成本低；

② 适合做长期保存、快速读写的场合。

(3) 光盘：高密度光盘使用的是光学存储介质而非磁性载体，它是用聚焦的氢离子激光束处理记录介质的方法存储和再生信息的一种数据存储设备。其特点如下。

① 寻址访问，保存简单，可靠性高，低成本。

② 适合做长期的数据保留且对写速度要求不高的场合。

(4) 磁带机(Tape Drive)：磁带机是传统数据存储备份中最常见的一种存储设备。磁带机一般指单驱动器产品，通常由磁带驱动器和磁带构成，是一种经济、可靠、容量大、速度快的备份设备。这种产品采用高纠错能力编码技术和写后即读通道技术，可以大大提高数据备份的可靠性，如图 1-5 所示。

(5) 磁带库(Tape Library)：磁带库是基于磁带的备份系统，磁带库由多个驱动器、多个插槽及机械手臂组成，并可由机械手臂自动实现磁带的拆卸和装填。它能够提供与磁带机相同的基本自动备份和数据恢复功能，但同时具有更先进的技术特点。它可以多个驱动器并行工作，也可以几个驱动器服务于不同的服务器来做备份，存储容量高达到 PB(1PB=1 000 000GB)级，可实现连续备份、自动搜索磁带等功能，并可以在管理软件的支持下实现智能恢复、实时监控和统计，是集中式网络备份的主要设备。磁带库不仅数据存储量大很多，而且在备份效率和人工占用方面拥有无可比拟的优势，如图 1-6 所示。



图 1-5　磁带机



图 1-6　磁带库

(6) 磁盘阵列(Disk Array)：磁盘阵列由一个或者多个磁盘子系统(通常是可访问的)中的磁盘组成的磁盘集合，这些磁盘由控制软件组合到一起并统一控制。控制软件将磁盘集合的总磁盘存储容量作为一个或者多个虚拟磁盘提供给主机。控制软件在磁盘控制器中运行的控制软件通常称为固件(Firmware)或者微码(Microcode)。在主机中运行的控制软件通常称为卷管理器(Volume Manager)。磁盘阵列通常由一个或者多个控制框级联一个或者多个扩展框构成，可以为应用系统提供高可靠性、大容量的数据存储空间，如图 1-7 所示。

(7) 虚拟磁带库(Virtual Tape Library，VTL)：虚拟磁带库集成了仿真软件的基于磁盘的备份系统，仿真软件可使基于磁盘的系统发挥磁带库的作用。这使得用户几乎不需要更改就能利用现有的备份与恢复过程和软件，同时提高了备份与恢复性能，可满足用户的恢

复时间和恢复点目标要求。虚拟磁带库允许使用现有的磁带备份软件，这使得管理人员使用磁带机做备份管理的经验可以被延续。VTL 由三部分组件构成：计算机硬件、应用软件(用于仿真磁带库和磁带驱动器)以及磁盘阵列。VTL 允许客户配置虚拟磁带驱动器、虚拟磁带盒和指定磁带盒容量。与物理磁带库不同，物理磁带库需要购买并安装额外的磁带驱动器，但对 VTL 来说，通过改变软件配置即可增加虚拟磁带驱动器，而这不需要花费任何额外的硬件成本，如图 1-8 所示。



图 1-7　磁盘阵列

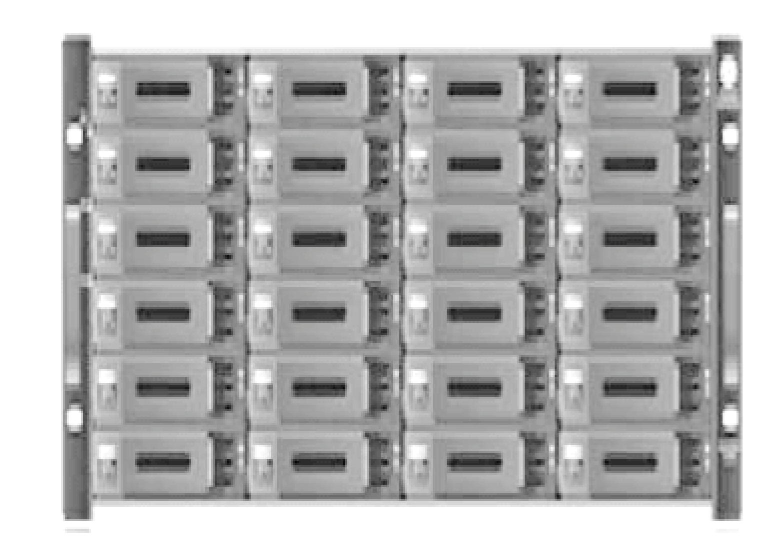


图 1-8　虚拟磁带库

1.2.3　存储网络的分类

由于计算机技术不断向更便宜、更有效的方向发展，早期的主机式计算机也从大型的中心式系统演化为便捷的、企业级的服务器。同时，网络技术对计算机平台的演化产生了相应的影响。随着这两项技术的逐渐成熟，以及对计算机处理能力和相关数据需求的不断增长，更好的存储技术将得到更多的市场驱动，存储网络也由此而生。

在过去的 10～15 年中，商业的模式发生了重大的改变，这其中，基于互联网的商业应用的爆炸性增长给信息的获取和存储技术带来了新的挑战。不断增长的对存储能力的需求使许多 IT 组织不堪重负，因此，发展一种具有低成本、高效益的先进存储方式就成为必然。

1. DAS

直接连接存储(Direct Attached Storage，DAS)是指将存储设备通过 SCSI 线缆或光纤通道直接连接到服务器上。随着用户数量的不断增长，尤其是用户数达到数百吉字节(GB)以上时，其在备份、恢复、扩展、灾备等方面的问题开始困扰系统管理员。

早期的数据存储方式大都是采用以硬盘为主要的存储媒体，对于网络上的文件共享及资料的存取，都需要通过文件服务器来完成，这种数据存储架构称之为直接连接存储架构。这种架构当初发展的目的就是希望将数据资源共享给网络上的使用者，但这种方式的主要缺点在于目前使用的文件服务器都需要通过某种常用的操作系统来达到资源共享的目的，而通常的操作系统的设计是为了多功能用途而规划的，并不是只针对数据的 I/O 部分去做最佳化处理，因此文件服务器这个角色常常会因为不必要的驱动程序或服务占据了系统资源，而导致文件存取的效能下降。

由于早期的网络以及应用非常简单，所以 DAS 存储架构被广泛应用。随着计算能力、内存、存储密度和网络带宽的进一步增长，越来越多的数据被存储在个人计算机和工作站中。分布式的计算和存储的增长对存储技术提出了更高的要求。由于使用 DAS，存储设备与主机的操作系统紧密相连，数据以及存储空间的共享存在较大的限制。同时，服务器系统

也因此背上了沉重的负担，因为 CPU 必须同时完成磁盘存取和进程运行的双重任务，所以不利于 CPU 指令周期的优化，典型 DAS 组网图如图 1-9 所示。

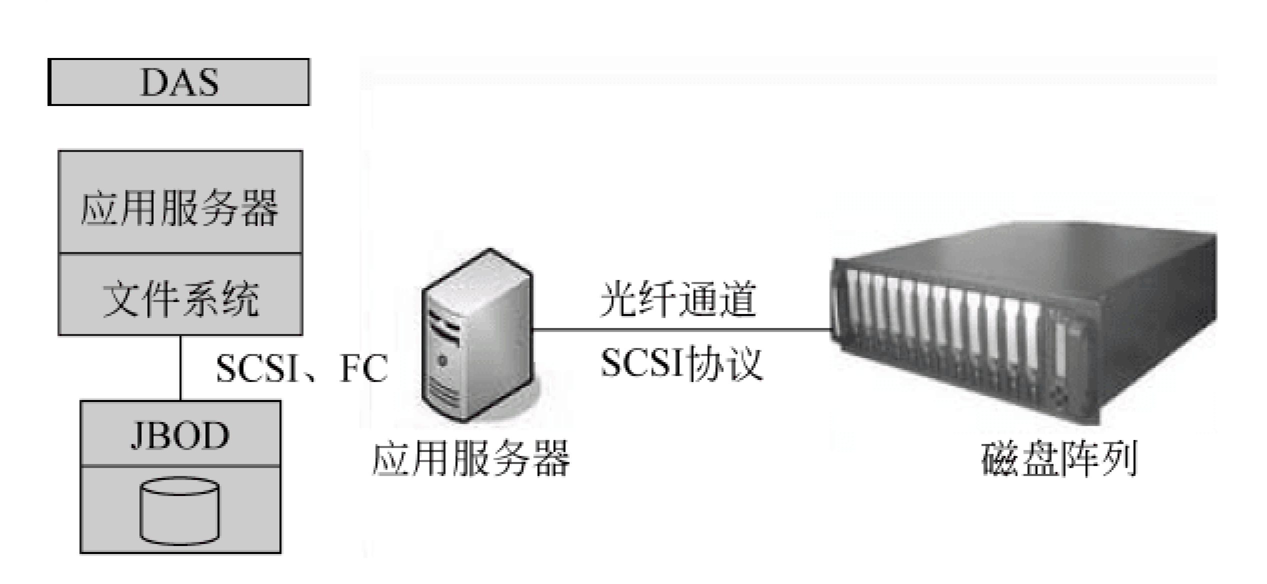


图 1-9　典型 DAS 组网图

DAS 具有以下特点：

(1) 存储设备(RAID 系统、磁带机和磁带库、光盘库)直接连接到服务器；

(2) 其使用传统的、常见的连接方式，容易理解、规划和实施；

(3) 没有独立的操作系统，不能提供跨平台的文件共享，不同平台下的数据需分别存储；

(4) 各个 DAS 系统之间没有连接，数据只能分散管理；备份软件需要服务器支持。

2. NAS

网络连接存储(Network Attached Storage，NAS)是一种文件共享服务。NAS 拥有自己的文件系统，通过 NFS 或 CIFS 协议对外提供文件访问服务，因此能使不同的操作系统进行文件共享。NAS 从结构上分为文件服务器和后端存储系统两大部分。文件服务器上装有专门的操作系统，通常是定制的 UNIX、Linux 操作系统，或者是一个简化的 Windows 系统。这些操作系统为文件系统管理和访问做了专门的优化。文件服务器(FS)利用 NFS 或 CIFS 协议，对外提供文件级的访问，因此 NAS 文件服务器也称 NAS 网关。后端存储系统主要由磁盘阵列构成，提供数据存储的空间支持，另外文件服务器的操作系统也直接集成在磁盘阵列上。

随着商业需求的增加，局域网技术得到广泛地实施，在多个文件服务器之间实现了互联，为实现数据共享而建立了一个统一的结构。随着计算机节点的增加，因系统平台不兼容而导致数据的获取日趋复杂。因此采用广泛使用的局域网加工作站的方法，对文件共享、互操作性和节约成本有很大的意义。

为了解决扩展及性能的问题，NAS 架构应运而生，这是一种直接通过现有业务网站链接的方式以提供不同的系统平台间进行文件共享的存储设备。其设计理念主要是做成一个专门负责文件 I/O 处理的高效能文件存储设备，将不必要的服务程序、工具软件全部进行整合，并且针对文件 I/O 的存取功能做了最佳化的处理，使得对文件存取的效率上较传统的文件服务器大为提升。

NAS 包括一个特殊的文件服务器和存储设备，NAS 服务器上采用优化的文件系统，并且安装有预配置的存储设备。由于 NAS 是连接在局域网上的，所以客户端可以通过网络与 NAS 系统的存储设备交换数据。另外，NAS 提供对多种网络文件传输协议的应用支持，诸如 NFS、CIFS 等，客户端系统可以通过磁盘映射与数据源建立虚拟连接。

对网络上的使用者而言，NAS 就像是一个大型的文件服务器一样，NAS 设备以文件共享设备的形态在网络上出现，NAS 是一种使用传统的以太网作为传输介质的存储装置，用户将所需共享的文件集中存放在 NAS 设备上，利用标准的网络传输协议(例如 TCP/IP)来与网络上的服务器或者客户机通信，并将存储空间共享给网络上的服务器或客户机使用。由于文件的集中存放，这使得共享文件的控制和管理更加容易，并可提升 IT 人员的管理效率。典型 NAS 组网图如图 1-10 所示。

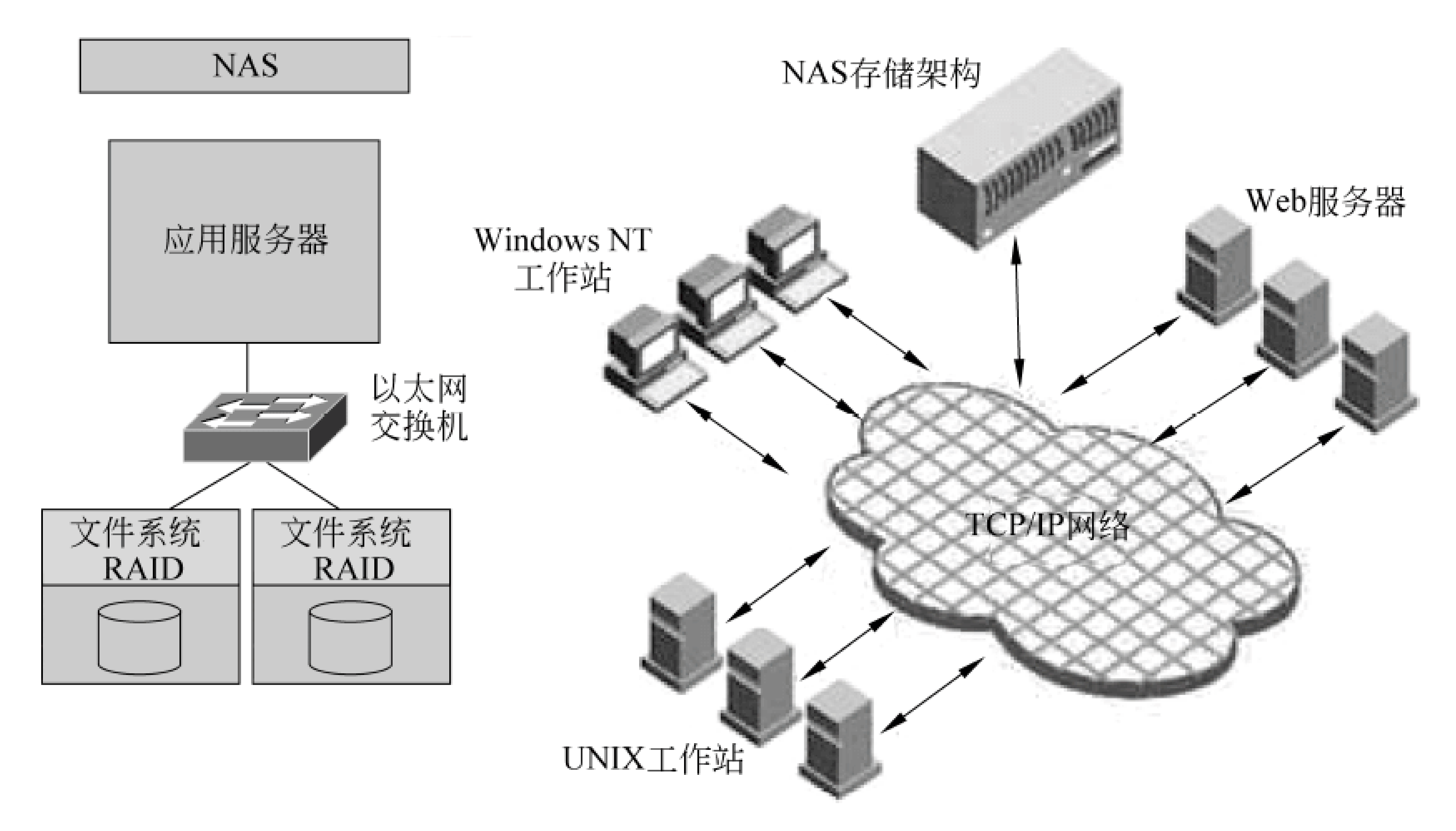


图 1-10　典型 NAS 组网图

NAS 具有以下特点。

(1) NAS 本身具有独立的操作系统，通过网络协议可以实现完全跨平台的文件共享。

(2) NAS 可以实现集中的数据管理，并且很多 NAS 产品都集成了本地的备份软件，可以实现无服务器备份功能。

(3) NAS 内每一个应用服务器通过网络共享协议(如 NFS CIFS)使用同一个文件管理系统。

(4) 磁盘 I/O 会占用业务网络带宽，同时 NAS 的性能也受到业务网络的影响。

3. SAN

存储区域网络(Storage Area Network，SAN)是一种通过网络方式连接存储设备和应用服务器的存储架构，这个网络专用于主机和存储设备之间的访问。当有数据的存取需求时，数据可以通过存储区域网络在服务器和后台存储设备之间高速传输。目前常用的 SAN 结构根据协议和连接器的不同，主要可以分为两种：一种是 FC SAN；另一种是 IP SAN。目前主流存储厂商的 FC SAN 的数据传输率已经达到 8Gbps。

这是一种用在服务器与存储资源之间的、专用的、高性能的网络体系，它为了实现大量原始数据的传输而进行了专门的优化。SAN 是一个存储网络架构，其主要作用是将服务器与存储设备分开，然后利用高速的光纤或者 IP 网络将二者连接在一起，从而使服务器可将其数据处理任务完全移交给存储装置完成，而服务器只需要专注于用户事务工作，然后再利用光纤通道或者 IP 网络来传输数据，以达到服务器与存储装置之间高效、稳定的存储环境。

构架 SAN 使用的典型协议组是光纤通道协议(Fiber Channel，FC)，在使用 FC 协议构

建的 SAN 中,FC 承载 SCSI 指令和数据,并为其提供更高的传输效率、更远的传输距离以及更好的传输质量。SAN 的应用主要集中在高端企业级的存储应用上,这些应用通常对于性能、冗余度和数据的可获得性都有很高的要求。

就应用而言,NAS 可视为一个以产品为导向的小型企业文件架构的解决方案,而 SAN 则是为大中型数据存储而规划与建设的存储架构解决方案。典型的 SAN 组网图如图 1-11 所示。

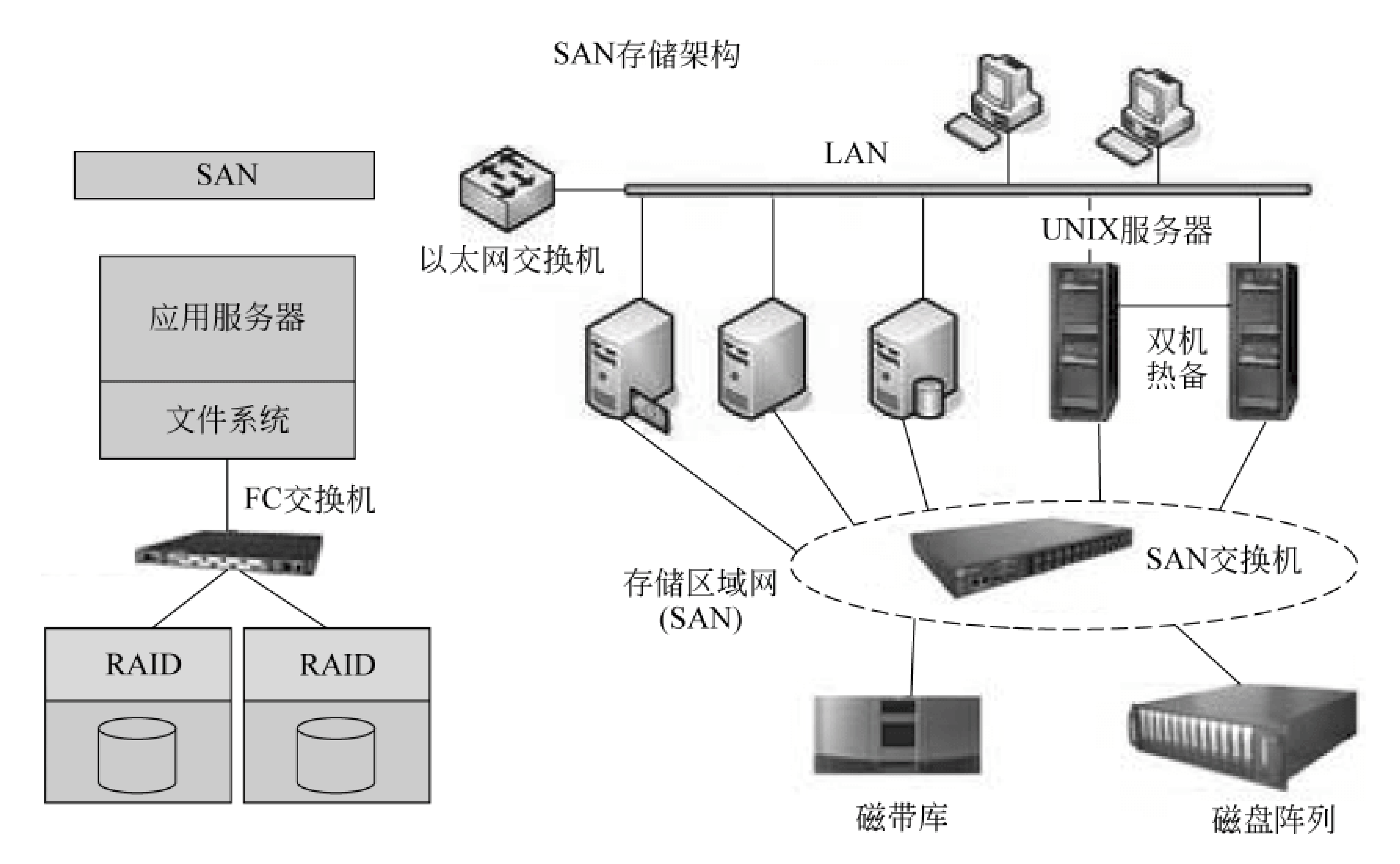


图 1-11 典型的 SAN 组网图

1.2.4 存储网络的比较

目前来看,每一种存储形态都会有自己的优点和缺点,在做出存储方案之前需要权衡一下利弊。至少有三个比较全面的存储选项值得考虑,即直连存储(DAS)、网络直连存储(NAS)和存储区域网络(SAN)。

1. 直连存储

接触过服务器的人都会对 DAS 比较熟悉。DAS 是一种将存储介质直接安装在服务器上或者安装在服务器外的存储方式,将存储介质连接到服务器的外部 SCSI 通道上也可以认为是一种直连存储方式。

由于这种存储方式在磁盘系统和服务器之间具有很快的传输速率,因此,虽然在一些部门中一些新的 SAN 设备已经开始取代 DAS,在要求快速磁盘访问的情况下,DAS 仍然是一种理想的选择。在 DAS 环境中,运行大多数的应用程序都不会存在问题,所以用户没有必要担心应用程序问题,从而可以将注意力集中在其他可能会导致问题的领域。

以下情况下,可以选择 DAS 方式存储。

(1) 如果你的存储系统中需要快速访问,但是公司目前还不能接受最新的 SAN 技术的价格时,或者 SAN 技术在用户的公司中还不是一种必要的技术时。

(2) 对于那些对成本非常敏感的客户来说,在很长一段时间内,DAS 仍然是一种比较便宜的存储机制。当然,这是在只考虑硬件物理介质成本的情况下才有这种结论。如果与

其他的技术进行一个全面的比较,考虑管理开销和存储效率等方面的因素,则 DAS 将不再占有绝对的优势。

(3) 对于那些非常小的不再需要其他存储介质的环境来说。

2. 网络直连存储

如实际应用中需要增加额外容量的时候,可以很容易地扩展 NAS 设备。市场上的一些 NAS 设备可以扩展到 200TB 的容量。在那些需要对数据进行块级访问的情况中,对数据库存储和 Exchange 信息存储来说,使用 NAS 方式更适合。

以下情况下,可以选择 NAS 方式存储:

在文件级访问系统中,数据的访问是通过文件名来实现的,因为文件名是带有一定含义的。而在块级访问系统中,数据的访问是通过数据块的地址来实现的,这个地址是特定数据存放的位置。在一个客户机/服务器的环境中,如果需要从文件服务器读取一个文件时,则要指定文件、服务器来完成数据块的读取工作,并且将得到的数据返回给用户就可以了。数据库存储和 Exchange 存储在这种方式的通信过程中存在着很多问题,所以它们并不适合存储于 NAS 设备中。使用 DAS 和 SAN 解决方案中提供的块级访问可以更为有效地实现数据库存储和交换存储中的数据。

虽然在需要将存储空间放在网络上时,NAS 是一个非常好的解决方案,但是 NAS 还有如下不足:

(1) 在拥有相同的存储空间时,它的成本比 DAS 要高很多;

(2) 对于数据库存储和 Exchange 存储这种使用率要求高的操作来说不是很适合;

(3) 获得数据的最大速率受到连接到 NAS 的网络速率的限制;

(4) 在存储基础设施中存在潜在的节点故障的可能。

3. 存储区域网络

在为存储解决方案中,SAN 是最昂贵的存储应用,同时也是最复杂的应用。虽然 SAN 在初始阶段需要投入大量的费用,但是 SAN 却可以提供其他解决方案所不能提供的能力,并且在合适的情形下可以为公司节约一定的资金。

SAN 解决方案通常会采取以下两种形式:光纤信道以及 iSCSI 或者基于 IP 的 SAN。光纤信道是 SAN 解决方案中最常用的类型,基于 iSCSI 的 SAN 解决方案开始大量出现在市场上,与光纤通道技术相比较而言,这种技术不仅具有良好的性能,而且价格低廉。

SAN 真正综合了 DAS 和 NAS 两种存储解决方案的优势。例如,在一个很好的 SAN 解决方案实现中,可以得到一个完全冗余的存储网络,这个存储网络具有不同寻常的扩展性,确切地说,可以得到只有 NAS 存储解决方案才能得到的几百万亿字节(TB)的存储空间,但是还可以得到块级数据访问功能,而这些功能只能在 DAS 解决方案中才能得到。对于数据访问来说,还可以得到一个合理的速度,对于那些要求大量磁盘访问的操作来说,SAN 显然具有更好的性能。利用 SAN 解决方案,还可以实现存储的集中管理,从而能够充分利用那些处于空闲状态的空间。更有优势的一点是,在某些实现中,甚至可以配置没有内部存储空间的服务器,要求所有的系统都直接从 SAN(只能在光纤通道模式下实现)引导。这也是一种即插即用技术。

SAN 确实具有很多的优点,当然 SAN 还有较大的缺陷,即成本大和复杂性,特别是在光纤信道中这些缺陷尤其明显。使用光纤信道的情况下,合理的成本是 1TB 或者 2TB,需

要五万到六万美金。从另一个角度来看,虽然新推出的基于 iSCSI 的 SAN 解决方案只需要两万到三万美金,但是其性能却无法和光纤信道相比较。在价格上的差别主要是由于 iSCSI 技术使用的是现在已经大量生产的吉比特以太网硬件,而光纤通道技术要求特定的价格昂贵的设备。

作为不同类型的存储解决方案,能够有一个正确的方向来快速做出存储解决方案十分必要。可以做出一个可以进行快速比较的图表,通过这个图表,可以比较不同类型的存储解决方案的优缺点。在这个图表中,SAN 分成 iSCSI 和光纤通道两种类型,以帮助用户区分这两种技术的不同。如表 1-1 所示。

表 1-1　不同存储形态的比较

存储形态	DAS	NAS	iSCSI/IP SANs	光纤通道
价格	价格较低	价格中等	价格中等到较高	价格较高
可扩展性	非常有限	依赖于解决方案	依赖于解决方案	依赖于解决方案
可管理性	效率较低	效率较低	非常高效	非常高效
容错性	较好	较好	很好	很好
是否适合文件存储	是	是	是	是
是否适合数据库存储	是	否	通常适合	是
是否适合网页服务	是	是	是	是
是否适合 Exchange	是	否	通常适合	是
安装的简易性	简单	简单	有一定的困难	非常困难
灾难恢复的能力	没有	没有	很多	很多
操作系统的支持	全部	N/A	Windows、Linux、UNIX、NetWare(其他系统是否支持依赖于驱动器本身)	Windows、Linux、UNIX、NetWare(其他系统是否支持依赖于驱动器本身)
主要提供商	任何服务器提供商	IBM、Dell、HP、Network Appliance	LeftHand、EMC、HP、IBM、NetworkAppliance	IBM、EMC、HP、Network Appliance

计算机系统中普遍存在一个“二八”定律,这个定律也同样在数据存储系统中有效,根据大量的调查数据以及经验数据表明,存储设备中保存的数据只有 20%左右是被用户经常读取的,而有 80%左右的数据很少甚至不被用户访问。这意味着,对保存在存储设备中的数据,20%的数据应该能够为用户提供及时的访问而不需要提供高效的访问效率。根据这样的特性,为了最优化使用存储设备,对不同级别的数据分级存储,这就是数据分级管理(Hierarchiacl Storage Management,HSM)的来由。

备份是指存储设备上的数据定时或按一定策略复制到备份介质上,通常的备份介质是磁带。被备份的数据仍热保留在存储设备上,备份的主要目的是为了防止存储设备上的数据被误删除或者意外丢失。

归档是值将重要的数据转移至某种介质上长期保存,通常的归档介质是光盘和磁带。归档和备份最主要的区别就是被归档的数据在原存储设备上是不被保留的。

迁移是指将存储设备上的数据复制到二级存储设备上，在原存储设备上保留占位符并释放空间。迁移和归档一样，都可以释放原存储设备上的空间。归档和迁移的区别是，迁移会从原存储设备回迁到原存储设备上，而无须进行人工干预。不过，有的设备的数据迁移方式需要在访问时系统直接从二级存储设备上读取数据，而不会将其回迁。

数据分级存储示意图如图 1-12 所示。

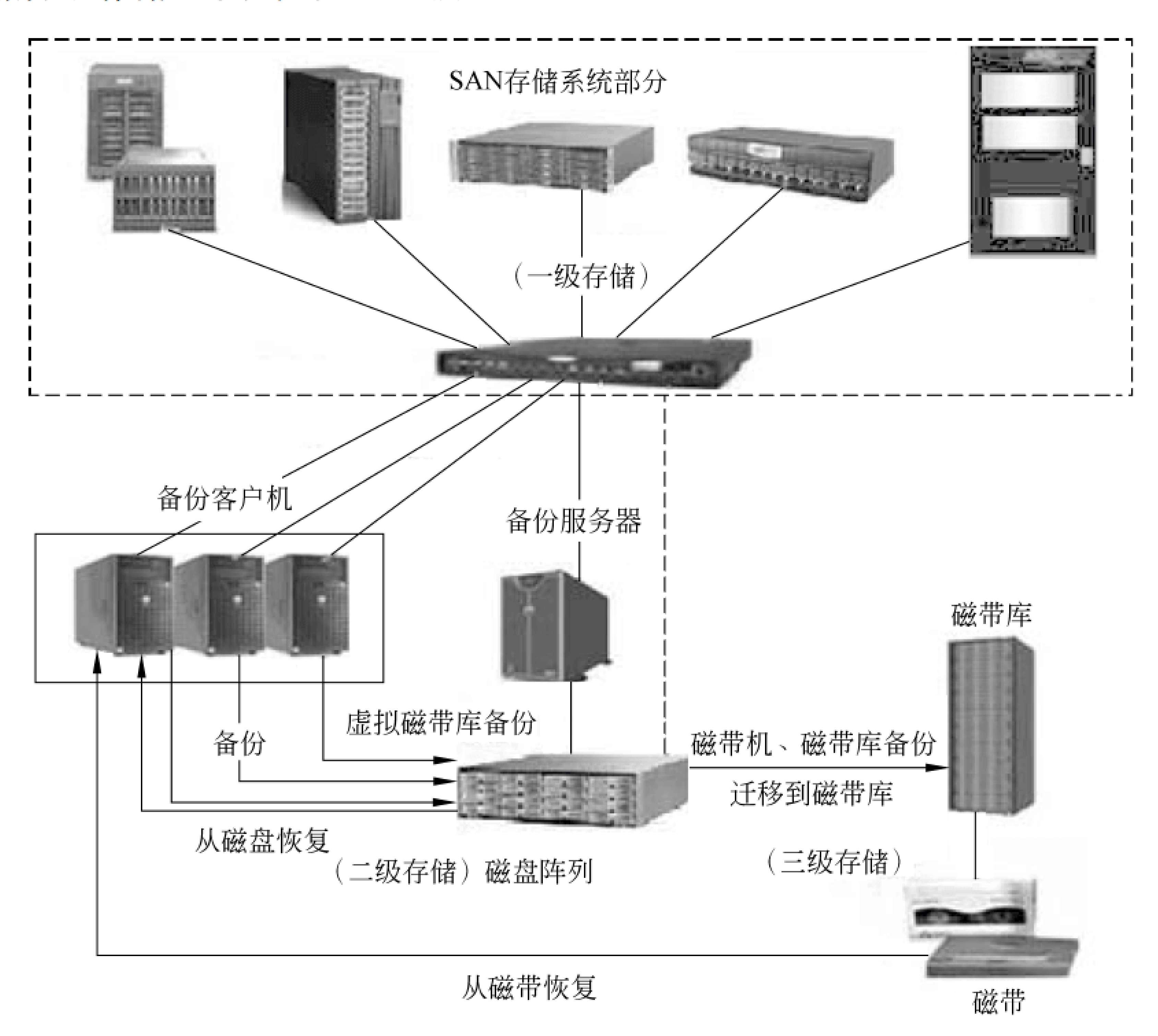


图 1-12　多级存储示意图

1.2.5　RAID 存储技术

传统的单盘容量和性能提升存在瓶颈，独立磁盘冗余阵列（Redundant Array of Independent Disks，RAID）是一种冗余磁盘阵列技术，该技术可以将多个磁盘组合为一个逻辑磁盘，从而突破单盘的容量限制，满足大数据存储空间的需求。同时 RAID 技术还可以提高磁盘的冗余度，传统机械磁盘难免会遇到物理故障，RAID 技术有效地解决了单盘故障造成数据丢失的风险。

常见的 RAID 级别有 RAID0、RAID1、RAID3、RAID5、RAID6 等。

1.2.6　存储性能指标

存储系统性能的三个主要衡量指标是最大带宽和输入/输出速率（IOPS）以及 SPC 测试报告。带宽通常也被称为数据吞吐量，通常用 MB/s 表示，表明最大持续不变的数据速率。通常最大的数据速率可以通过读或写操作的顺序数据流检测，数据块的大小为 64KB

或更大。输入/输出速率是系统每秒钟能够完成输入/输出(I/O)的最大值。最大输入/输出速率通常也是通过读或写操作的顺序数据流进行测量,数据块的代销为单一扇区的大小或者是512字节。存储性能理事会(Storage Performance Council,SPC)提供较权威的存储性能基准测试,包括SPC和SPC-1 IOPS。

IOPS(I/Os per second)表示即每秒输入输出次数,指的是系统在单位时间内能处理的最大I/O频度。一般情况下,联机事务处理系统(On-line Transations Processing,OLTP)应用涉及更多的频繁读写,更多情况下应考虑IOPS。

IOPS测试结果与很多测试参数和存储系统具体配置有关。IOPS还可以细分为100%顺序读IOPS、100%顺序写IOPS、100%随时读IOPS、100%随机写IOPS等,在同等情况下这四种IOPS中,100%顺序读的IOPS最高。

厂商公布的经常是IOPS很高的100%顺序读的指标,但多数用户实际实用的环境既有顺序读写,也有随机读写操作。传输的数据块大小也不相同,所以产品在用户实际使用环境中的性能通常会比厂商标称的指标差。

SPC的SPC-1基准测试主要是针对随机I/O应用环境的,SPC-2基准测试主要是针对顺序I/O应用环境的。SPC-2基准比SPC-1基准具有更高的权威性和可比性。

SPC-1基准测试虽然规定了严格的读写顺序和随机读写比例和数据块大小以及在何种磁盘负载情况下取值,但没有规定被测存储产品使用多少个磁盘,也没有规定被测存储产品设置何种RAID级别。存储性能理事会(SPC)要求测试报告必须详细地列出被测存储系统的配置和价格。

SPC网站(http://www.storageperformance.org)上公布了大多数存储厂商的存储产品的SPC-1基准测试报告,通过查询这些报告中的SPC-1 IOPS值和所描述的被测存储产品的磁盘总数,用户就可以根据下列公式快速估算所采购配置的某一存储实际性能为

$$\text{实际 SPC-1 IOPS}=(\text{实际采购存储系统的磁盘数}\div\text{测试报告中被测系统的磁盘数})\times\text{测试报告的 SPC-1 IOPS}$$

可靠性为:

$$\text{MTT Farray}=\text{MTT Fdisk}\div\text{实际采购存储系统的磁盘数}$$

1.3 数据存储技术

1.3.1 备份技术背景

随着数据存储技术在商业系统中的普及以及大量应用系统的上线,企业信息安全的重要性日益凸显。但作为信息安全的一个重要内容,数据备份的重要性往往被忽视。只要发生数据传输、数据存储和数据交换,就有可能产生数据故障,而一些自然灾害和人为的错误也在威胁着信息的安全,这些情况都可能造成数据丢失、数据被篡改甚至系统瘫痪等后果,而作为系统管理员则必须要维护数据的完整性和准确性,以保证系统和业务的持续运行。

在信息系统中,备份是指为原始数据制作一个或者多个额外的拷贝并将其存放在其他存储介质中,以便在原始数据受到破坏或者特定情况下加以重新利用的一个过程。数据备

份的目的主要有两个：其一是在灾难发生后用于恢复原始数据状态，这也被称为灾难恢复；其二是用于数据的恢复，即当原始数据文件在被意外删除或损坏的情况下恢复原始数据。由此可见，数据备份的根本目的不在于数据的重新利用。也就是说，备份工作的核心是备份恢复，一个无法恢复的备份，对任何系统来说都是毫无意义的。能够安全、方便又高效地恢复数据，才是备份系统的真正意义。对一个完整的 IT 系统而言，备份工作是其中必不可少的组成部分，其意义不仅在于防范意外事件的破坏，同时也是归档及保存历史数据的最佳方式。

1.3.2 数据备份原则

1. 稳定性

备份产品的主要作用是为系统提供一个数据保护的方法，所以备份系统的稳定性和可靠性就是非常重要的一个因素。备份软件通常被要求要与操作系统 100%的兼容，并且当事故发生时能够快速有效地恢复数据。

2. 全面性

在复杂的应用环境中，应用系统可能采用了多种操作平台，包括 UNIX、Windows、Linux 等，并安装了各种应用系统，如 ERP、数据库等。而备份系统要求能够支持各种操作系统、数据库和典型应用，以满足复杂的实际应用需求。

3. 自动化

很多系统由于工作性质的原因，对何时备份、用多长时间备份都有一定的限制。在非工作时间且系统负荷较轻时较适合备份。因此，备份方案应能提供定时的自动备份，并利用自动磁带库等技术自动更换磁带。在系统备份过程中还要有日志记录功能，并在出现异常情况时自动报警。

4. 高性能

随着业务的不断发展，数据越来越多，更新越来越快，在休息时间来不及备份如此多的内容，所以需要考虑提高数据备份的速度，并采用多种技术加快对数据的备份，同时充分利用通道的带宽和性能。

5. 操作简单

数据备份到磁盘(Data to Disk，D2D)方式正逐渐被越来越多的用户采用，其基本数据流程为：备份服务器按照既定策略，在相应时间内发出控制命令，将生产服务器主存储磁盘的数据通过 LAN 或 SAN 备份到相应的磁盘设备中，通常可用于数据的分级存储。

数据备份到磁盘后再备份到磁带(Data to Disk to Tape，D2D2T)方式结合了传统磁带的离线管理和磁盘高速备份恢复的特性，基本数据流程为：备份服务器按照既定策略，在相应时间内发出控制命令，将生产服务器主盘的数据通过 LAN 或 SAN 备份到相应的次级磁盘存储设备中(如：虚拟磁盘库)，再由相应生产主机备份服务器在既定时间内自动将保存在次级磁盘存储设备中的数据复制到磁带库中，这样可以大大缩短对备份窗口的需求，并能够有效减少对应用系统资源的占用，使得备份效率得到极大的提升。

1.3.3 数据灾难基础

已出台的《重要信息系统灾难恢复规则指南》中有明确定义：“灾难是由于人为或者自

然的原因,造成信息系统运行严重故障或瘫痪,使信息系统支持的业务功能停顿或服务水平不可接受,导致信息系统需要切换到备用场地运行。”由此可见,灾难不仅指客观的原因,也包括人为的因素。在信息系统中,一切能导致系统非正常停机的时间都可以称为灾难。大致可以分成以下四个类型。

(1) 自然灾害:包括地震、洪水、雷电等,这种灾难破坏性大,影响面广。

(2) 社会灾难:包括战争、火灾、盗窃等。

(3) IT 系统灾难:包括主机的 CPU、硬盘等损坏,电源中断以及网络故障等,这类灾难影响范围比较小,破坏性也小。

(4) 人为灾难:包括黑客攻击、病毒侵入、误操作、蓄意破坏等。

1.3.4 数据容灾与备份

容灾,就是当灾难发生时,保证生产系统的数据尽可能少地丢失,并保持生产系统的业务不间断地运行。

备份是容灾的基础,是指为了防止系统遭受人为的误操作或者其他故障而导致数据丢失,而采取的将全部或者部分数据从应用主机的存储设备复制到其他存储设备的过程。我们把这种数据备份方式称为冷备份。

数据备份的核心是恢复,采取的措施主要有双机热备,磁盘镜像或容错、备份介质异地存放,关键部分冗余等多种灾难预防措施。这些措施能够在计算机发生单点故障后进行系统恢复,对于一些毁灭性的灾难不具有恢复能力。

数据容灾是指能够在灾难发生时全面、及时地恢复整个系统,避免传统冷备份的不足,例如国际标准 SHARE 78 定义的容灾系统有 7 个层次:从最简单的仅在本地进行磁带备份,到将备份的磁带存储在异地,再到建立应用系统实施切换到异地备份系统,以及容灾级别所对应的系统恢复时间也从几天到小时级、分钟级、秒级或零数据丢失等。

无论采取哪种容灾方案,基本手段都是数据备份,因为任何容灾方案都不可能脱离备份的数据而实现。衡量容灾系统的指标主要有两个:RPO(Recovery Point Object)和 RTO(Recovery Time Object),其中 RPO 代表了当灾难发生时丢失的数据量,而 RTP 则代表了恢复系统所需要的时间。

在建立容灾系统之前,首先要进行全面的需求分析,其中包括业务系统风险分析、容灾系统对业务系统的影响分析和成本分析。

(1) 风险分析:风险分析是检查哪些是可能造成数据损失或者系统瘫痪的外在和内在因素。既然是容灾,必须充分考虑业务系统所在的自然环境,针对可能发生的灾难,准备相应的容灾对策。

(2) 容灾系统对业务系统的影响分析:容灾系统肯定对业务系统的性能有一定影响,因此,对于那些高负荷运行的业务系统,必须认真计算。

(3) 成本分析:建立容灾系统,除了需要购买必要的设备外,还要考虑系统维护管理成本和使用通信线路的费用。这些容灾成本也是构建容灾系统所必须考虑的因素。

1.3.5　数据容灾指标

1. RTO

RTO(Recovery Time Objectives,恢复时间目标)是能够加快恢复数据存储和压缩正常运行的时间。一个 5 分钟的 RTO 是数据保护的决定性因素,而 RTO 决定数据恢复的时间。一个 5 分钟的 RTO 表明丢失的数据必须在 5 分钟内恢复出来并且能够正常使用。更进一步说,可以没有停顿地恢复数据,并且能够重新正常使用机器。RTO 需要考虑的一个因素是能够在一段特定的时间内恢复数据,同时还能恢复服务器操作系统以及安装相应的软件来使用相应的数据。例如,如果只是需要恢复服务器上的数据文件,那么同时还需要在服务器上恢复相应的操作系统和设备或安装另外的数据恢复产品。因此,RTO 需要考虑的因素有备份操作的完整性、数据的恢复、数据的重新存储和重启机器所需要的设备等。

RTO 示意图如图 1-13 所示。

关键业务决定了从中断点恢复到其最低业务持续目标(MBCO)所能承受的最大时间,从而使中断对业务所带来的冲击最小化。



图 1-13　RTO 示意图

2. RPO

RPO(Recovery Point Objectives,恢复点目标)是实时地复制业务信息中的每一个数据恢复事务。短时间的 RPO 能够更少地丢失数据。例如,一个 5 分钟的 RPO 表明必须在 5 分钟内恢复数据,而一小时的 RPO 表明这种数据恢复在一小时内可能已经丢失了要备份的数据。相反地,一个 0 分钟的 RPO 表明没有数据可以丢失,因为数据已经及时地备份或者记录下来,从而阻止任何数据的丢失,RPO 要考虑的另外一个层面是数据的保护要完整和全面到什么程度,例如,RPO 如果每隔 24 小时备份一次,则意味着这 24 个小时内数据可能会丢失,完全和全面的数据保护注重的是数据是否完整地被保护起来或者只有部分的文件和数据被保护起来。再举一例,打开的文件可能无法被完全备份,除非缓存中的数据存储到了磁盘里。另外还要考虑的因素是索要备份的文件是否是某个特殊的目录或者文件共享中的某种特定文件,以及数据是否完全备份下来了。小的 RPO 意味着要付出更多的费用以及更少的数据丢失量,应用时必须做一个权衡。PRO 示意图如图 1-14 所示。

1.3.6　数据容灾级别

根据 SHARE 78 国际组织提出的标准,容难恢复解决方案可分为七级,即从低到高有七种不同层次的灾难恢复解决方案。可以根据企业数据的重要性以及业务所需要恢复的速度和程度来设计、选择并实现业务的灾难恢复计划,如图 1-15 所示。

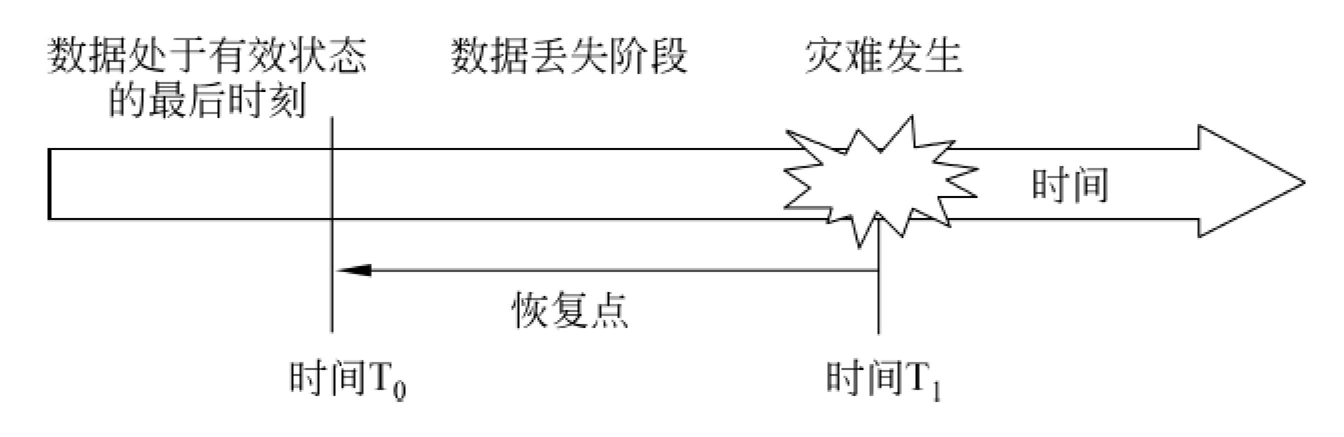


图 1-14　PRO 示意图

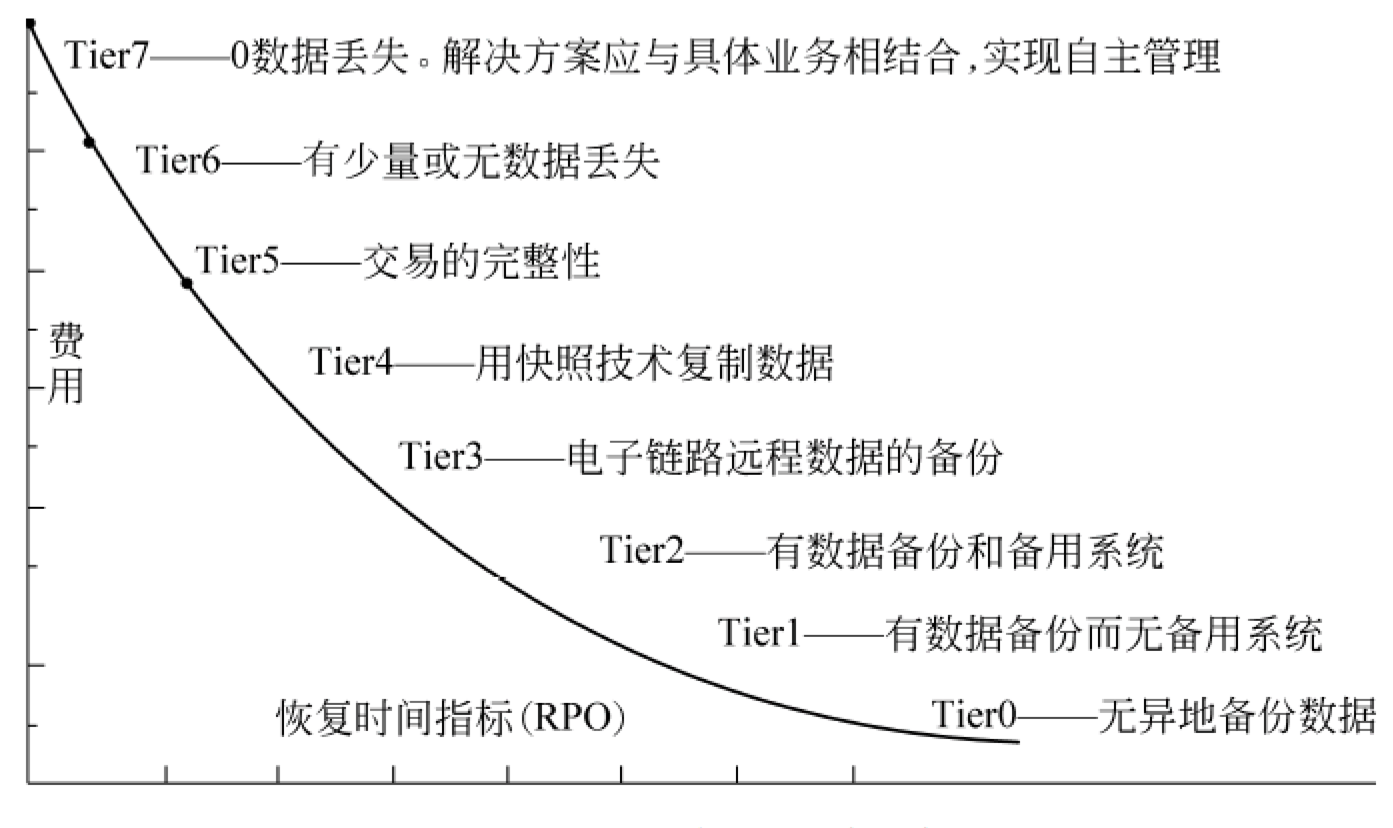


图 1-15　七级容灾级别示意图

（1）1 级：本地保存

Tier1 表示没有任何异地备份或应急计划。即数据仅在本地进行备份恢复，没有数据送往异地。事实上这一层并不具备真正灾难恢复的能力。

（2）2 级：异地保存

Tier2 的灾难恢复方案必须设计一个应急方案，能够备份所需要的信息并将它存储在异地。PTAM 指将本地备份的数据用交通工具送到远方。这种方案相对来说成本较低，但难于管理。

（3）3 级：网络传输

Tier3 相当于 Tier1 再加上热备份中心能力的进一步的灾难恢复。热备份中心拥有足够的硬件和网络设备区来支持关键应用。其相对于 Tier1 明显降低了容灾恢复时间。

（4）4 级：自动备份

Tier4 是在 Tier2 的基础上用电子链路取代了数据传送的更进一步的灾难恢复方式。由于热备份中心要保持持续运行，增加了成本，但提高了灾难恢复的速度。

（5）5 级：采用中间件

Tier5 指两个中心同时处于活动状态并互相备份，在这种情况下，工作负载可能在两个中心之间分享。在灾难发生时，关键应用的恢复也可以降低到小时级或分钟级。

（6）6 级：数据级容灾

Tier6 则提供了更好的数据完整性和一致性。也就是说，Tier5 需要两中心与中心的数据都被同时更新。在灾难发生时，仅是传送中的数据被丢失，恢复时间被降低到分钟级。

(7) 7 级：应用级容灾

Tier7 可以实现无数据丢失，被认为是灾难恢复的最高级别，在本地和远程的所有数据被更新的同时，利用了双重在线存储和完全的网络切换能力。当发生灾难时，能够提供跨站点动态负载平衡和自动系统故障切换功能。

1.4.1　数据存储与数据访问

随着计算机在人们生活、工作中的普及，计算机存储系统作为计算机系统的组件也越来越受到重视。存储系统的发展随着计算机技术的发展而进入了存储网络阶段。从而向人们提供了更大的存储空间和更加便捷的存储访问能力。

互联网的范围已经从人们生活、工作、旅游度假等延伸到各个方面，如此庞大的数据信息保存在各个巨大的在线信息库中，从而实现对各种各样信息的访问。实际上，所有对数据的请求和访问，都是依赖于世界各地各种类型的服务器的。服务器对数据进行保存，并在人们需要这些数据时能提供准确、有效的访问。随着计算机的飞速发展，人们对数据信息的存取方式和可达性提出了更高的要求，而网络的出现为数据存取和访问方式的实现提供了强大的动力，使得人们无论身处何处，都能够方便、快捷地对数据进行访问。

数据量的不断增长和人们对数据访问性能要求的不断提升，推动了计算机存储技术的发展，而网络的出现和大规模应用级的存储技术变得日益普及。正因如此，人们现在所面临的问题已经由寻找足够的存储空间去保存数据转变为保存什么样的数据，如何保存这些数据以及将这些数据放在哪里的问题。一方面，人们所面对的数据量非常庞大；另一方面，人们需要在如此庞大的数据中能方便地访问所需要的数据，所以，网络技术和存储技术的融合是历史的必然。长期以来，爆炸式增长的数据的存储和大量的数据访问需求的矛盾一直是摆在存储技术人员和用户面前的一大难题。随着个人存储设备和企业数据中心对存储容量的要求和数据访问能力要求的不断提升，这个问题也越来越突出。

1.4.2　存储网络对应用系统的影响

应用系统中的非线性的性能扩展主要受到两方面因素的影响。首先是在线存储容量是否充足、可用，以满足应用系统数据的需求，并且应该有足够的临时存储资源，包括内存和高速缓存，以满足应用系统数据的要求，其次是与应用系统进行交互操作从而访问在线存储设备以及应用数据或者是向存储设备中写入新数据的用户量，应用系统在处理用户数据访问的时候应该能够利用临时在线存储资源以保证能够以接近实时的方式处理预定数量的事务。

在线存储的可用性问题中，如果用户将要与应用系统进行交互操作，那么与该交互操作相关的信息就应该能够实时地访问，而在线存储所提供的就是去满足这种要求的机制，并且，在线存储容量应该确保有足够的空间以保存当前用户的数据和应用操作所需的数据，同时还应当有足够的空闲容量，以便在尽可能不影响应用系统操作的情况下支持用户数据的拓展。

访问应用系统的用户数量对于应用系统性能拓展的影响,也可以被理解为规划的数据访问量。因此,配置方案中可以设定在可接受的响应时间窗口内所能够服务的最大用户数量。尽管这种优化配置还需要服务器运算能力和充足的网络资源的支持。当时,从根本上来说,应用系统所能够承受的用户失误需求取决于存储基础设施。

C/S 存储组网模式如图 1-16 所示。

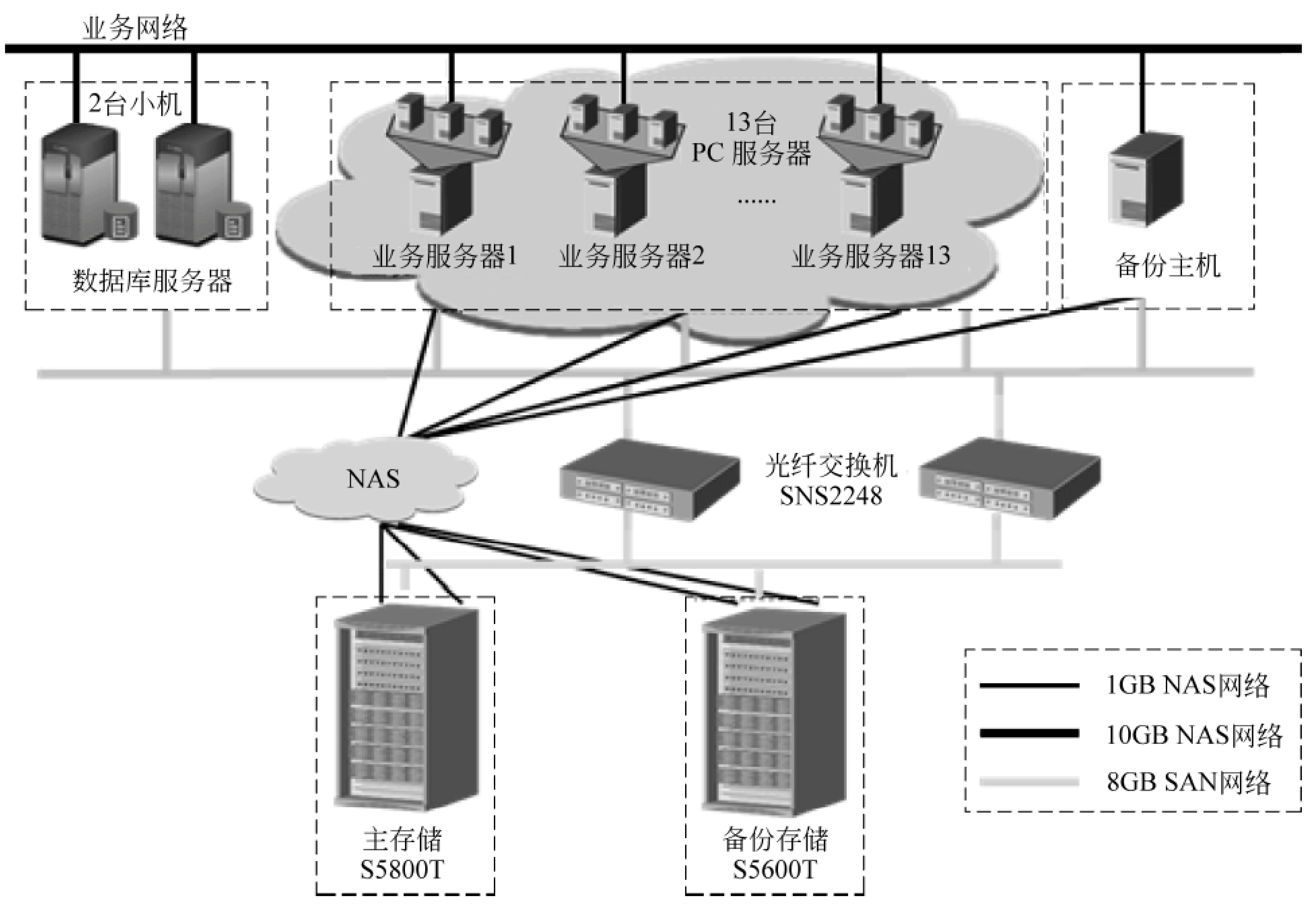


图 1-16　C/S 存储组网模式

现代存储配置方式体现的是分布式计算形式下的客户机/服务器模式的特征,客户机/服务器存储模式为服务器提供了数据存储的能力,也为客户端提供了数据存储的空间。

随着在线存储的增大,服务器的性能也越来越强大,并配置了更多的 RAM。由于客户端用户需求的增长速度高于服务器能力的提高速度,因而服务器很快就会达到其性能的极限。因为服务器的性能限制,最优化的存储配置几乎是不可能实现的。

除了处理来自客户端的访问请求以外,为了实现数据的查找,服务器的信息共享也提出了大量的存储容量需求,因此需要在网络中设置多台服务器以满足不断增长的数据信息的存储需求,由此导致了服务器的专门化演变。网络服务器专门负责处理与客户机登录网络相关的各种工作,保存网络参数信息,并管理网络资源,而客户端资料以及共享的信息则保存在各自的文件服务器上。对在线存储空间的需求以及客户端的多重访问,要求网络中部署多态服务器以共同承担负载。同时,数据库的规模也越来越大,这就要求在网络中部署数据库服务器。数据库服务器的出现,为客户机/服务器存储模式起到了重要的推进作用,并促使其成为一种新的、拓展的存储解决方案。

在传统的 C/S 模式中,存在着一些难以克服的局限。随着数据和在线存储要求的不断增加,系统的容量和功能都必须不断地提升以应对所面临的挑战。尽管服务器中各种临时

需求量总是在无限攀升，而服务器的性能却存在提升的极限。并且，在 C/S 体系架构中都有可能会收到用户的访问，在各个应用数据日益壮大的今天，越来越多的用户通过互联网与应用系统相连接并且每个人都需要存储越来越多的数据，这种在线工作环境所带来的数据传输需求是一种极大的挑战。

数据流量所带来的挑战是导致存储技术变革的直接因素之一。如今，数据存储以及数据访问所面临的问题也已经促使存储技术突破，激发出创造性的解决方案来应对 C/S 存储模式的局限性。存储网络的出现改变了传统的存储方式，将存储"直接"与服务器相连变成与"网络"相连。这种设计方式直接将存储设备接在网络上，将存储连接从服务器分离出来，从而动态地改变了服务器的 I/O 能力，为彻底解决应用系统性能扩展问题奠定基础。同时，在这一变革的基础上，人们将能够构建扩展性极高的存储基础设施，用以处理大规模的数据访问任务，在服务器之间共享数据，并提高大规模在线存储来提高管理效率。

存储网络建立了独立的存储架构，增强了现有 C/S 模式的计算拓扑结构，并从整体上改变了 C/S 模型。存储网络允许存储设备直接连接到现有的业务网络上（比如 NAS），也可以通过专门存储网络进行连接（比如 SAN），存储网络的应用为传统的存储配置方案带来了两个重要的变化。

（1）存储网络在存储设备、服务器以及客户端之间建立了更多的直接访问路径，通过冗余路径以及负载均衡的应用，为应用系统提供更加高效、可靠的数据存储访问环境。

（2）存储网络的应用可以为商业应用系统提供更高的访问数据效率，将服务器的性能释放出来，使服务器能够更专注于用户事务的处理，提升了服务器的服务性能。

存储网络根据应用环境的不同而发展出了网络连接存储（Network Attached Storage，NAS）和存储区域网络（Storage Area Network，SAN）。NAS 可以将存储设备直接连接在基于以太网标准的现有业务网络中，并可以使用标准的 TCP/IP 网络协议与服务器或者是客户机进行通信。而 SAN 则专门为存储建立一个基于光纤通道协议或者 IP 协议的独立网络，以保证服务器系统能够与存储系统间独享高带宽。

1.4.3　存储网络对应用系统的意义

存储网络的出现，使服务器能同时连接和利用更多的存储设备，从而消除因对单个服务器的大量访问而产生的性能瓶颈，同时也不需要为每台服务器维护一个数据副本，且因其所带来的复杂性和开销都将不复存在。从应用系统的角度看，这一转变使得应用数据可以分布在所有的服务器上，在保持对相关数据的集中访问的前提下，避免了应用中单点故障的存在。

存储网络的部署所带来的带宽提升为大量数据的传输提供明显优势，通过 FC 建立的 SAN 或者是通过以太网络架构的 NAS 系统都能对日益增长的应用需求提供支持，并为维护稳定、可靠的响应时间奠定基础。

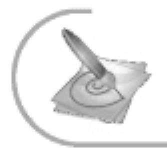
对于平台环境的维护需要有多重维护/支持应用系统的配合，而在 SAN 环境中，各种设备之间得以连接从而具备相互通信的能力，这样的能力使得许多基于服务器的、以数据为中心所维护/支持的应用系统能够得到优化。支撑应用系统可以从应用服务器上分离出来，这意味着数据可以在存储网络设备上直接进行复制。以往的许多应用系统一直被这样的维护工作所困扰，因为数据在复制过程中是不可用的。但是新的操作方式（存储网络中，数据

直接在存储设备之间复制)可以大幅度地缩短数据不可用时间,从而提高整个应用系统的可用性。

任务拓展

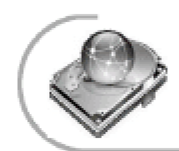
1. 查阅最近几年相关数据灾难事件及影响。
2. 数据容灾级别及相关指标。
3. 了解数据存储技术对目前应用系统的影响。

第 2 章　存储应用环境



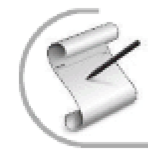
任务目标

- 了解 IT 基础设施数据中心；
- 了解数据存储的应用环境；
- 了解存储硬盘的物理结构；
- 了解存储介质的实现原理；
- 了解物理主机的应用环境。



项目背景

21 世纪是数字化和多媒体化的信息时代，现代信息社会和经济的发展，所产生的信息量每年以指数方式上升，出现了信息爆炸的态势。应用数据每天以成倍的速度增长。据 IDC 监测，全球在 2010 年已经进入数据量的 ZB(相当于 1 万亿 GB)时代，预计到 2020 年，全球将总共拥有 35ZB 的数据量，相对于 2010 年，数据量将增长近 30 倍。信息存储技术作为信息技术的核心之一，一直伴随并推动着 IT 业各方面技术的协同发展，是当今 IT 领域中少数发展最为迅速的热点之一。如何安全、合理地保存这些新增数据成为信息技术发展的必经之路。



项目描述

随着“互联网＋”、大数据时代的到来，聚比特科技有限公司随着业务和人员的扩大，已由一家单一电子商务逐步发展成为致力于推动国民经济及社会可持续发展的综合 IT 运营服务机构，逐步实现“资源全球化、经营本土化”的战略发展阶段。

以此建立一套完善的网络基础平台作为信息化支持中心、应用推广中心和公司技术管理中心，进一步适应今后业务的充分扩展，以保证公司业务的顺利开展，创建良好的办公自动化与业务系统数据中心。



项目分析

聚比特科技有限公司数据中心机房在符合国家相关规范(例如防火消防规范等)前提下，以布局的合理、可靠、安全和工作流程舒适性为布局设计原则。计划建设项目总建筑面积约为 178 平方米，使用面积约为 123 平方米。分为三个功能区域，分别为主设备机房、动力机房、操作间、钢瓶间。各间需要单独隔开。隔开后主设备机房用于放置配线柜、机柜、服务器、小型机、网络设备、通信设备等重要设备；动力机房放置 UPS、电池、配电柜等。



项目实现

数据中心机房属于大型重要的计算机中心，为保证计算机机房安全、可靠地运行，以及为工作人员提供一个良好的工作环境，数据中心机房建设项目严格按照国家设备选型及设计规划标准执行实施，符合当前企业 IT 系统要求，建设具备一个完整的中心机房工程系统。

2.1 IT 基础设施

2.1.1 计算机系统组成

企业 IT 系统离不开计算机/服务器，计算机系统的构成可以划分为四个层面，即计算资源、内部总线、存储资源和软件资源。计算资源主要由计算的核心硬件 CPU 来实现。大家知道，CPU 运算速度越快，计算机的计算能力也越强。数据临时存储由内存来完成，而长久存储则由内置硬盘、光盘、软盘等来实现。计算机内部的数据通信则由其内部总线(BUS)来完成，包括数据的输入、输出等。一个完整的计算系统包括硬件及与之相匹配的软件。在系统软件的基础上，为实现特定的业务功能应用，针对应用软件的开发也必不可少。

由此可以看出，在独立的计算系统中，数据计算、存储通常依靠 CPU 和内置的存储设备来实现，在计算要求越来越快、计算规模和需存储数据量也越来越大时很容易出现瓶颈。如图 2-1 所示为计算机系统的组成。

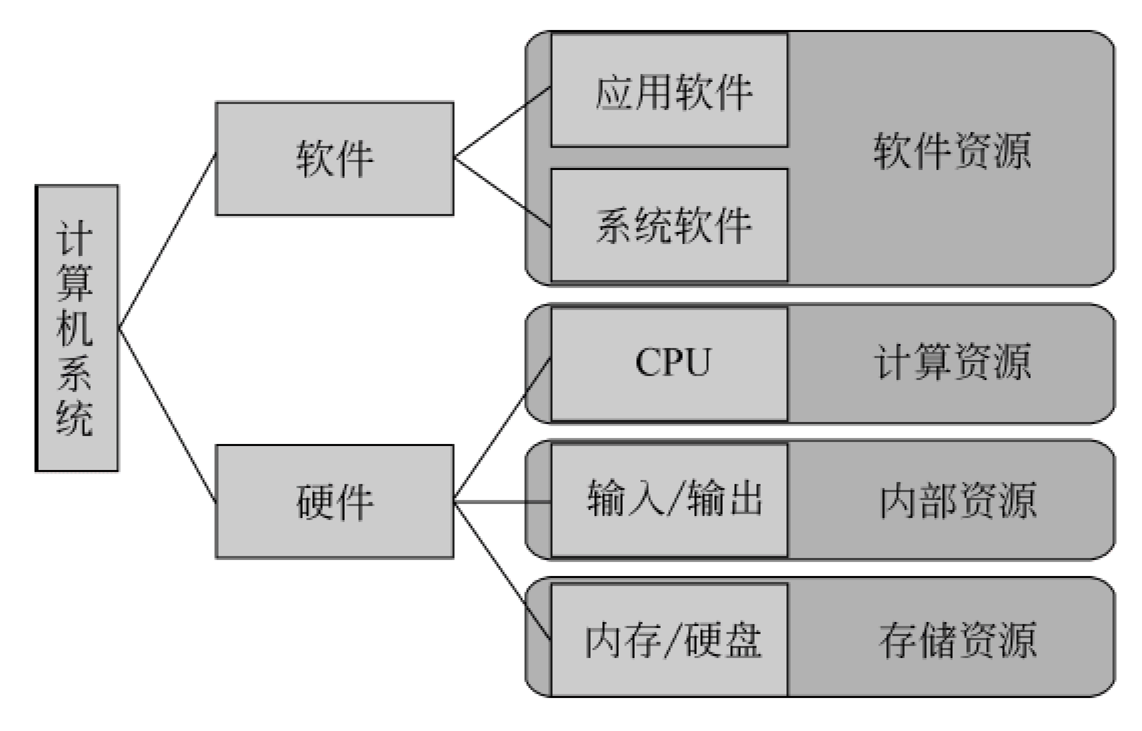


图 2-1　计算机系统的组成

2.1.2 IT 系统简介

1. IT 系统组成

随着大型计算、海量数据存储的发展，对计算能力、数据存储资源方面都有更高的要求，独立的计算机系统已经很难满足这些要求。因此，就需要把多个计算机系统集成起来，构成了一个整体的 IT 系统。IT 系统就是在计算机系统的基础上所进行的扩展和延伸，从架构上仍然可以将其划分为计算资源、存储资源、网络资源和软件资源四个部分。

(1) 计算资源：在 IT 系统中，往往会把多台服务器组成集群，通过集群方式实现计算

资源的负载均衡，提升整体计算能力，同时提高系统的冗余，保证系统的可靠性。

（2）存储资源：在 IT 系统中，存储资源从内置存储向外扩展成为外部存储，通过构建专用的外部存储系统，数据存储得到很大的扩展，为大数据量的存储提供了必备条件，同时，保证了数据的安全可靠性。

（3）网络资源：从独立的计算机系统发展成为 IT 系统，必然需要强大的网络资源提供数据通路，计算机系统内部总线已经不能满足 IT 系统网络资源的要求，因此基于 TCP/IP 网络和基于 FC 协议的 FC 网络架构得到长足的发展，已经成为 IT 系统网络资源的主流。

（4）软件资源：在 IT 系统架构中，软件资源不仅仅是独立计算系统中的单一操作系统，而是发展成为集群软件系统、分布式文件系统等，通过这些方式实现集群业务管理和分布式应用，如图 2-2 所示。

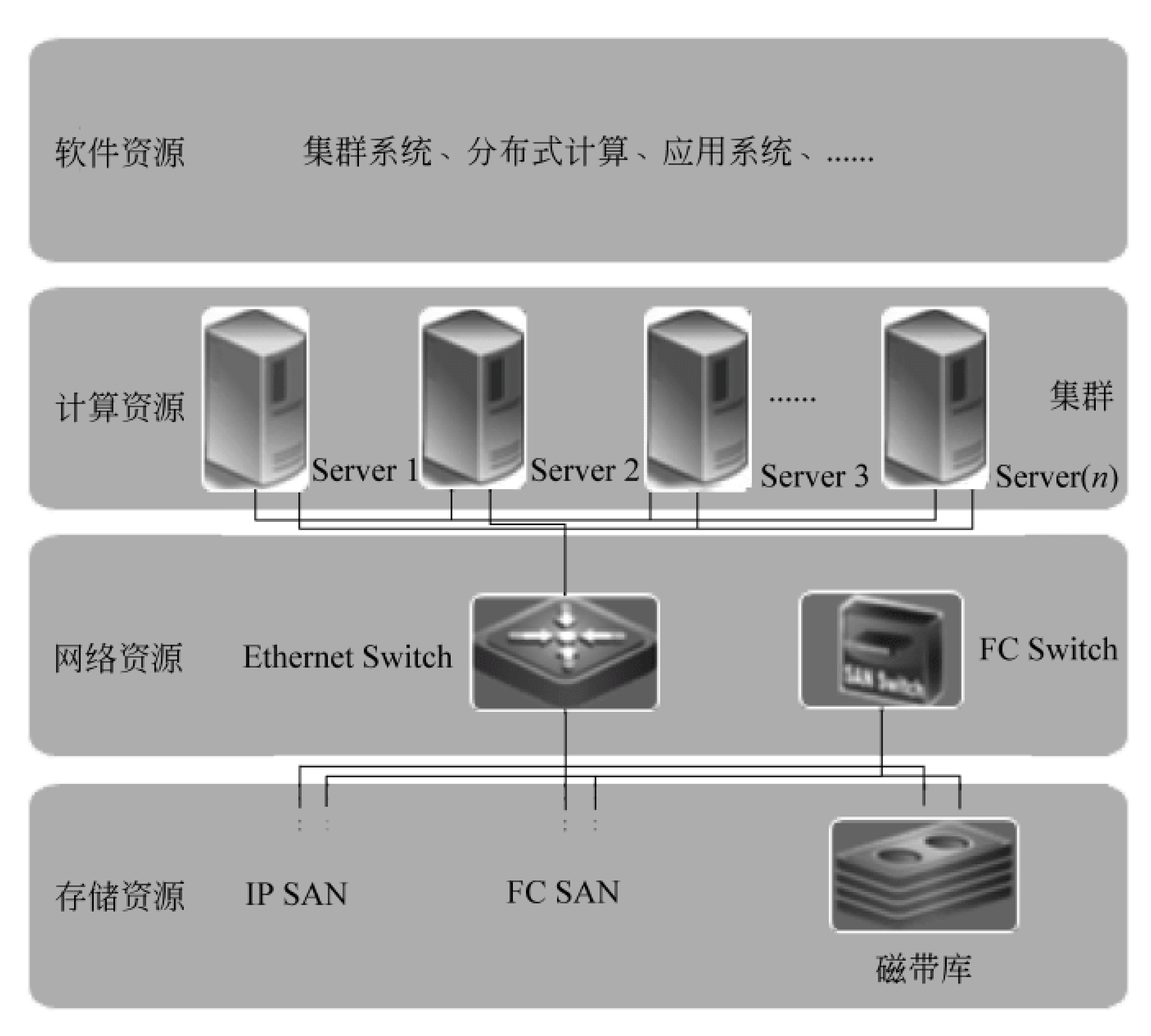


图 2-2　IT 系统

2. IT 系统软件

在 IT 基础设施中，软件也是必不可少的一部分。软件可以从下到上分为：硬件底层驱动、操作系统、数据库、应用软件。

硬件底层驱动和应用软件之间需要实现相应的信息交互。一方面，应用程序通过对驱动程序发送指令，实现硬件控制的动作指令；另一方面，驱动程序将从硬件上获得的数据传送给应用程序，实现应用程序与驱动程序间的交互。也就是说，硬件底层驱动实现了访问底层硬件的人机交互。例如：主板芯片驱动、网卡驱动等。

操作系统是管理计算机硬件与软件资源的计算机程序，提供一个让用户与系统交互的操作界面。操作系统需要处理如管理与配置内存、决定系统资源供需的优先次序、控制输入与输出 I/O 设备、操作网络与管理文件系统等基本事务。例如：微软 Windows 操作系统、

Linux 操作系统、UNIX 操作系统、AIX 系统等。

数据库(Database)是按照数据结构来组织、存储和管理数据的仓库。随着信息技术和市场的发展,数据库又发展出很多种类型,从最简单的数据表格到能够进行海量数据存储的大型数据库系统都在各个方面得到了广泛的应用,例如 Oracle、DB2 等。

应用软件是为满足用户不同领域、不同业务应用需求而提供的上层软件,它可以拓宽计算机系统的应用领域,放大硬件的功能。例如 E-mail 应用、财务系统等,如图 2-3 所示。

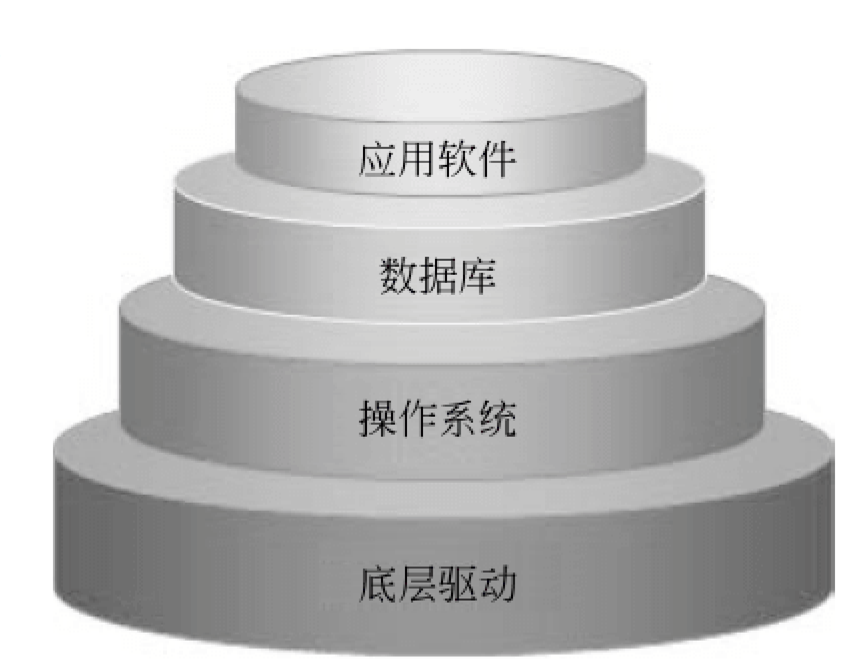


图 2-3　IT 系统软件

3. IT 系统计算资源

IT 系统中的计算资源通常由各式各样的服务器构成。当单个的服务器计算能力不能够满足应用需求的时候,采用服务器集群的形式提供计算能力,如图 2-4 所示。



图 2-4　服务器集群

4. IT 系统网络部分

在 IT 系统基础设施网络架构中主要划分成 4 个层次,从下到上依次为:存储层、服务器层、核心层、外部接入层。

存储层通过 TCP/IP 或 FC 网络连接到服务器层,为服务器提供数据存储空间资源,服务器层接入核心层,内部和外部用户通过接入层、汇聚层连接核心层,在核心层实现快速数据交换。为实现远程容灾,企业需要建立异地灾备中心,通过专用网络实现存储层与灾备中心的互联。

5. IT 系统存储资源

存储资源根据存储的位置可分为内部存储和外部存储,后面重点介绍外部存储,如 SAN 存储和 NAS 设备,如图 2-5 所示。

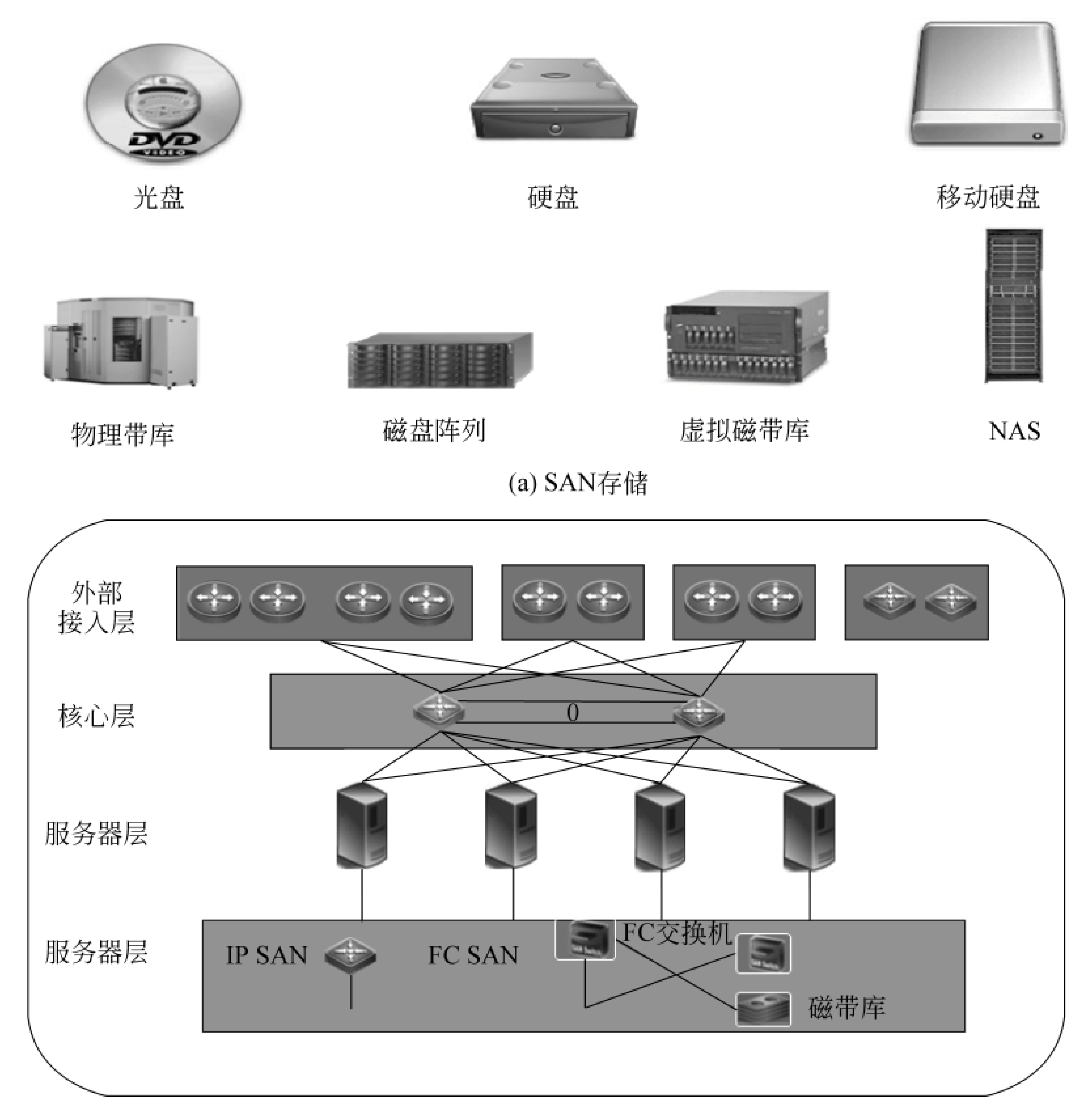
6. 传统 IT 系统面临的问题

(1) 资源利用率低:服务器资源、网络资源、存储资源难以得到有效的利用。

(2) 系统可靠性难以保障:设备的可靠性,传输链路的可靠性难以保障。

(3) 统一管理难:每一个设备是单独的一套警告系统,要了解整个系统的情况,需要到每一台设备上去查看,每一套设备有单独的密码,管理人员要分别去记忆,设置复杂了难于记忆,设置简单了存在着一定的安全隐患。

基于上述原因,出现了数据中心,使以上的问题得到了很好的解决。



(a) SAN存储

(b) NAS存储

图 2-5　SAN 存储和 NAS 存储

2.1.3　数据中心的概念

数据中心(DataCenter)通常是指在一个物理空间内实现信息的集中处理、存储、传输、交换、管理。

计算机设备、服务器设备、网络设备、存储设备等通常认为是数据中心的关键设备。关键设备运行所需要的环境因素,如供电系统、制冷系统、机柜系统、消防系统、监控系统等通常被认为是关键物理基础设施,如图 2-6 所示。



图 2-6　数据中心

1. 数据中心结构

广义的数据中心是企业(机构)的业务系统与数据资源进行集中、集成、共享、分析的场地、工具、流程的有机组合。其核心内容包括业务系统、数据 ETL、ODS 数据库、数据仓库、数据集市、商务智能等,也包括物理的运行环境(中心机房)和运行维护管理服务。具体来说它包含以下四个方面。

(1) 数据中心提供所有应用系统(包括集中的业务应用系统、数据交换平台、应用集成平台)的运营环境。

(2) 数据中心是容纳用以支持应用系统运行的基础设施(包括机房、服务器、网络、存储设备)的物理场所。

(3) 数据中心包括数据中心本身的 ODS、数据仓库及建立在其上的决策分析应用。

(4) 数据中心有一套成熟的运行、维护体系支持其日常运行,保证应用系统高效、准确、不间断地运行。

根据数据中心的定义和发展趋势,可以将数据中心划分为以下 4 个层次。

基础设施层:用统一的技术将机房、通信、计算、存储等 IT 基础资源融合形成数据中心的基础设施,为业务系统提供基本的资源服务,提高资源利用率以及 IT 系统的可靠性。

基础软件层:信息资源是企业生产过程中所涉及的一切文件、资料、图表和数据等信息的总称。本层存储了企业(机构)生产和经营活动所产生、获取、处理、存储、传输和使用的一切信息资源。

管理调度:实现存储资源化、计算资源化、网络资源化,并能够动态调整资源匹配数据的读写存储,解决统一管理难的问题。

应用层:主要包括针对结构化和非结构化数据的各种应用,包括各种业务系统、辅助决策系统和各种多媒体应用(监控、流媒体、统一通信、呼叫中心、视频会议、VOIP),如图 2-7 所示。



图 2-7 数据中心结构

2. 存储在数据中心的功能

(1) 集中存储:分散的存储不便于集中管理,存储资源也不利于共享,造成资源的浪费。

(2) 海量空间：随着 Internet 的发展，海量的非结构化数据的诞生，对数据中心的存储也提出了新的要求，在一定程度上，海量存储空间的大小直接决定着数据中心的发展规模。

(3) 快速 I/O：由于数据中心的用户众多，I/O 的快速响应能力直接决定着用户的应用感知，因此，快速的 I/O 也为数据中心的高效运行提供了保障。

2.2　存 储 环 境

2.2.1　主机的内部应用环境

主机服务器大部分 I/O 开始于需要访问数据的应用，应用通常不考虑存储后端的操作细节，而是直接调用由操作系统提供的系统调用接口，然后由操作系统支持的文件系统为数据提供数据的逻辑地址和在磁盘上存储的物理地址的映射，再通过设备驱动层，主要是 SCSI 协议的操作，将数据存储到存储设备(比如主机服务器内部硬盘)上。因此主机服务器内部数据存储 I/O 流程中的各个环节对数据存储的可靠性、性能和安全性都起到非常重要的作用，从独立的主机服务器来看，其内部 I/O 流程各个环节即共同构成了存储的内部应用环境，如图 2-8 所示。

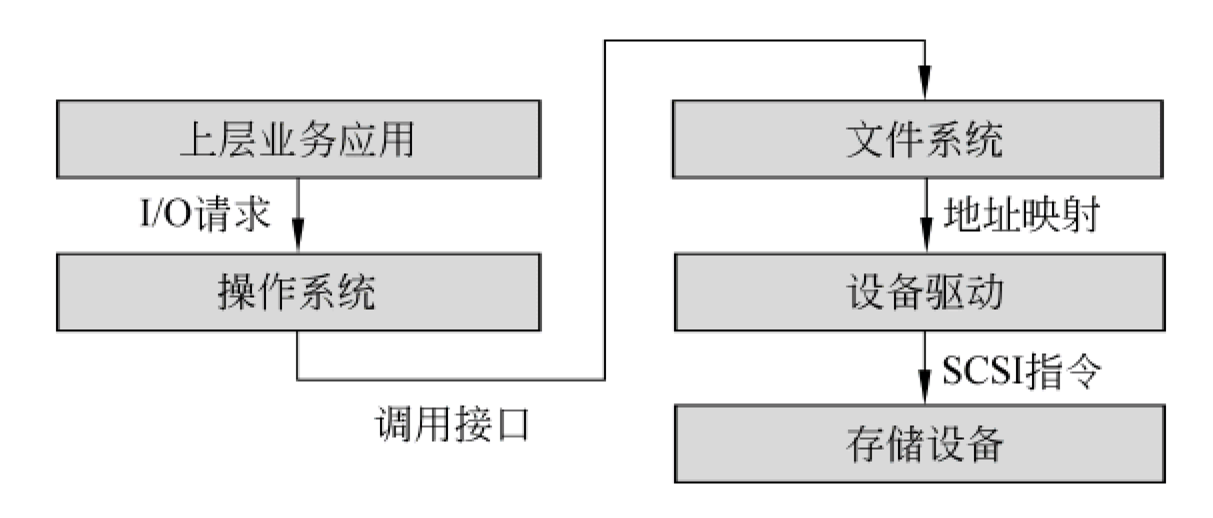


图 2-8　主机的内部应用环境

2.2.2　传统内置存储遇到的问题

在传统的计算机存储系统中，存储工作通常是由计算机内置的硬盘来完成，而采用这样的设计方式，硬盘本身的缺陷很容易成为整个系统的性能瓶颈，并且由于机箱内有限的空间，限制了硬盘数量的扩展，同时也对机箱内的散热、供电等提出了严峻的挑战。再加上不同的计算机各自为战，使用各自内置的硬盘，导致从总体看来存储空间的利用率较低，并且分散保存的数据也不利于数据的共享和备份工作。在传统的 C/S 架构中，无论使用的是何种协议，存储设备都直接与服务器相连接。在这样的结构下，对存储设备上所保存的所有数据的任何读写操作，都必须由服务器来进行，这样的处理方式给服务器带来了沉重的负担。

外部存储系统的出现，彻底将服务器从烦琐的 I/O 操作中解放出来，使服务器更加专门化，使之仅仅承担应用数据的操作任务，以更充分地释放自身潜能，如图 2-9 所示。

2.2.3　网络存储应用环境

网络存储系统按照不同的功能，可划分为三层。

第一层：主机层。运行相关应用，发起存储 IO 操作。在主机侧需要存储连接的设备主

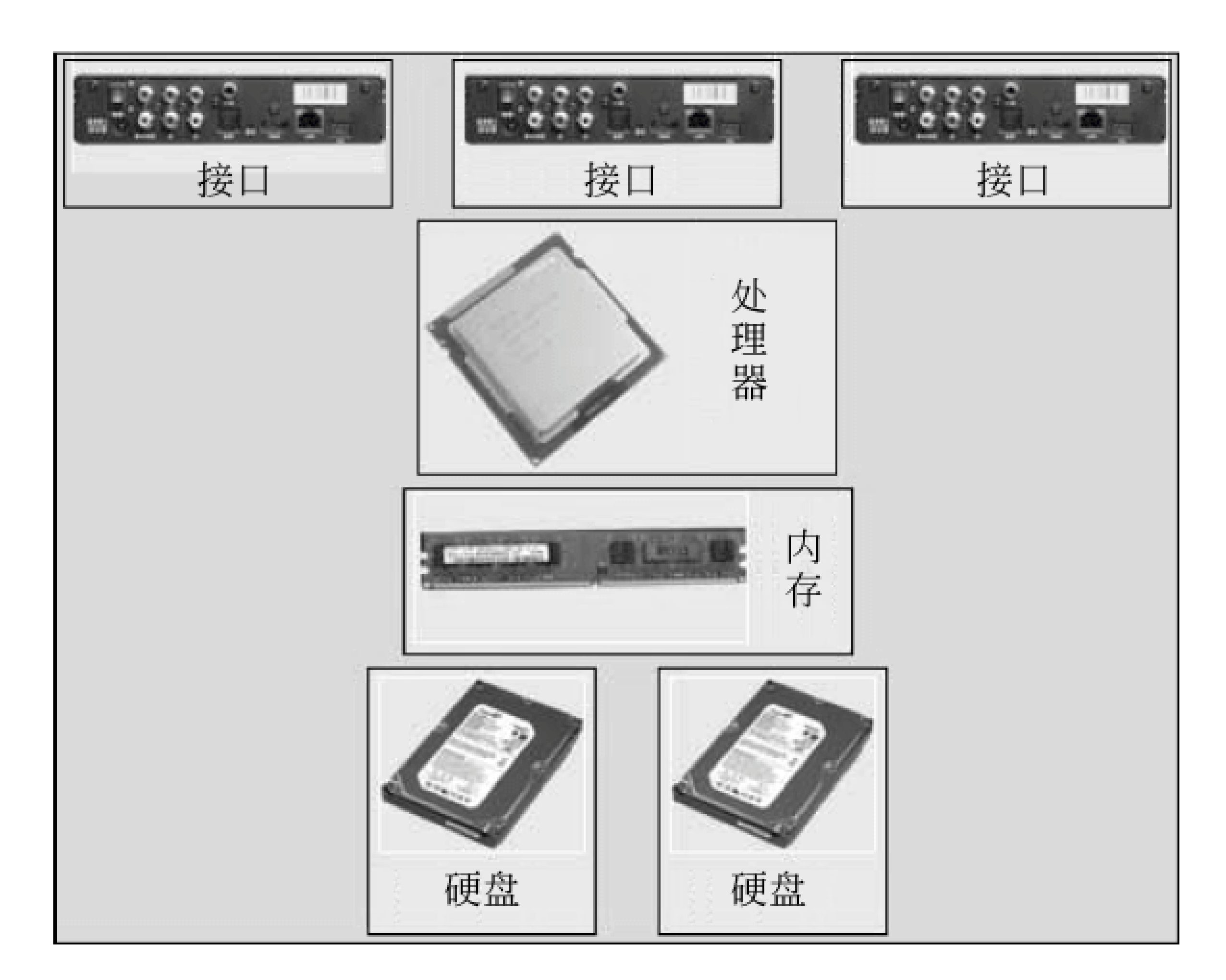


图 2-9　传统内置存储

要有 FC HBA 卡、iSCSI HBA 卡、以太网卡，需在主机侧安装的存储连接软件包括 initiator 启动器、open-iSCSI、多路径软件等。

第二层：网络层。将主机与存储层互联，提供网络通路，可采用 FC 网络或者以太网络的方式。以太网络可利用原有以太网络连接，利用现有资源组网。FC 网络需要建立专门的网络，速度快，但是 FC 器件都较昂贵，成本高。

第三层：存储层。存储系统的核心层，对数据进行存储、管理。

上面提到的每一层都将对数据存储的可靠性、性能、安全性产生重要影响，因此在进行存储系统建设时，需要从以上各层使用的硬件设备、软件协议、组网架构等方面考虑，以保证业务应用对数据存储可靠性、性能、安全性方面的需求，如图 2-10 所示。

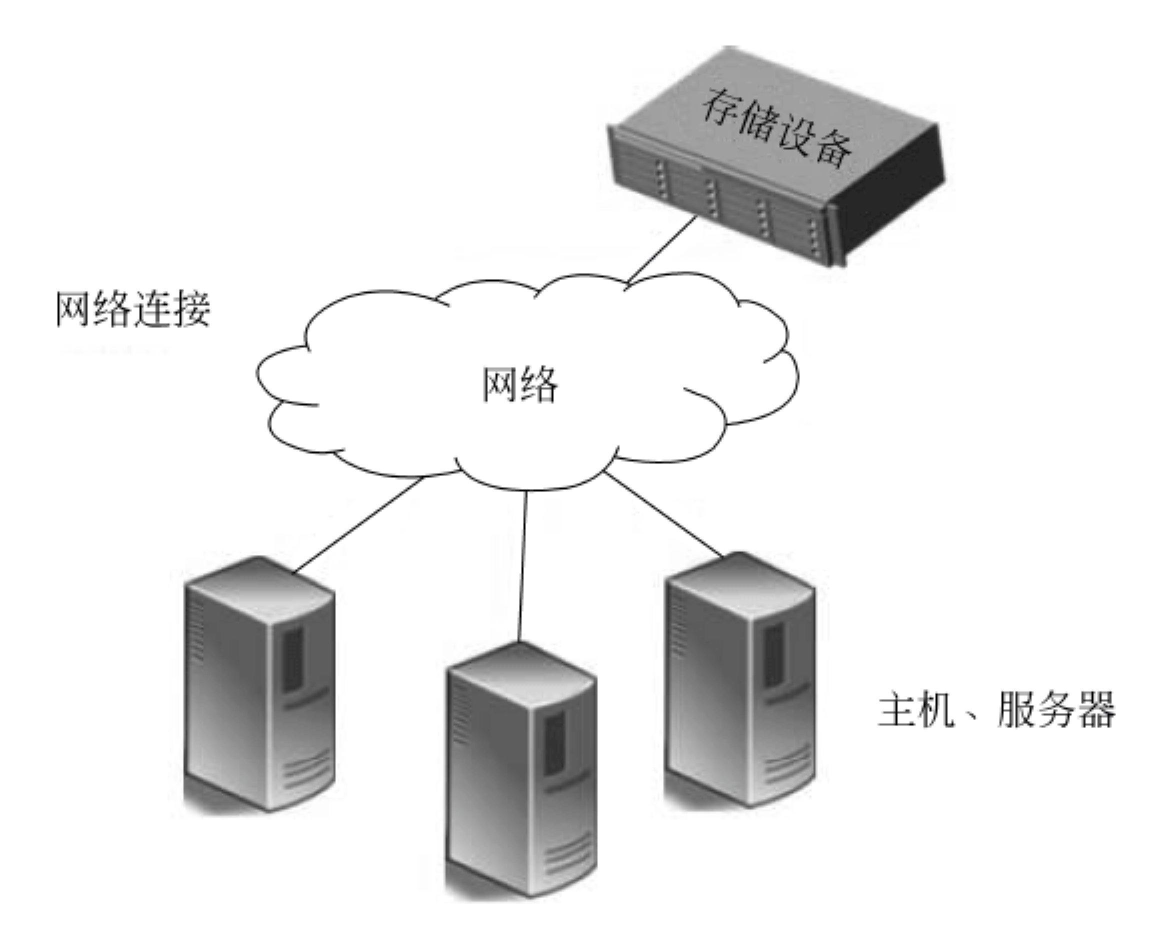


图 2-10　网络存储系统

2.3.1　机械硬盘

1. 机械硬盘的结构

机械硬盘包含机械装置和电子装置，如图 2-11 所示。可以分为如下部分。

(1) 磁头组件：用于数据的读取和写入。

(2) 磁头驱动机构：用于驱动磁头臂将磁头送达指定的位置。

(3) 盘片组：数据的载体。

(4) 主轴驱动装置：驱动盘片维持高速运转。

(5) 控制电路：系统控制、调速、驱动等。

(6) 接口：用于硬盘与主板连接，常见的接口类型有 ATA、SATA、SAS。



图 2-11　机械硬盘

2. 机械硬盘的磁头技术

机械硬盘的磁头技术如图 2-12 所示，主要包括机械部分和电子部分。

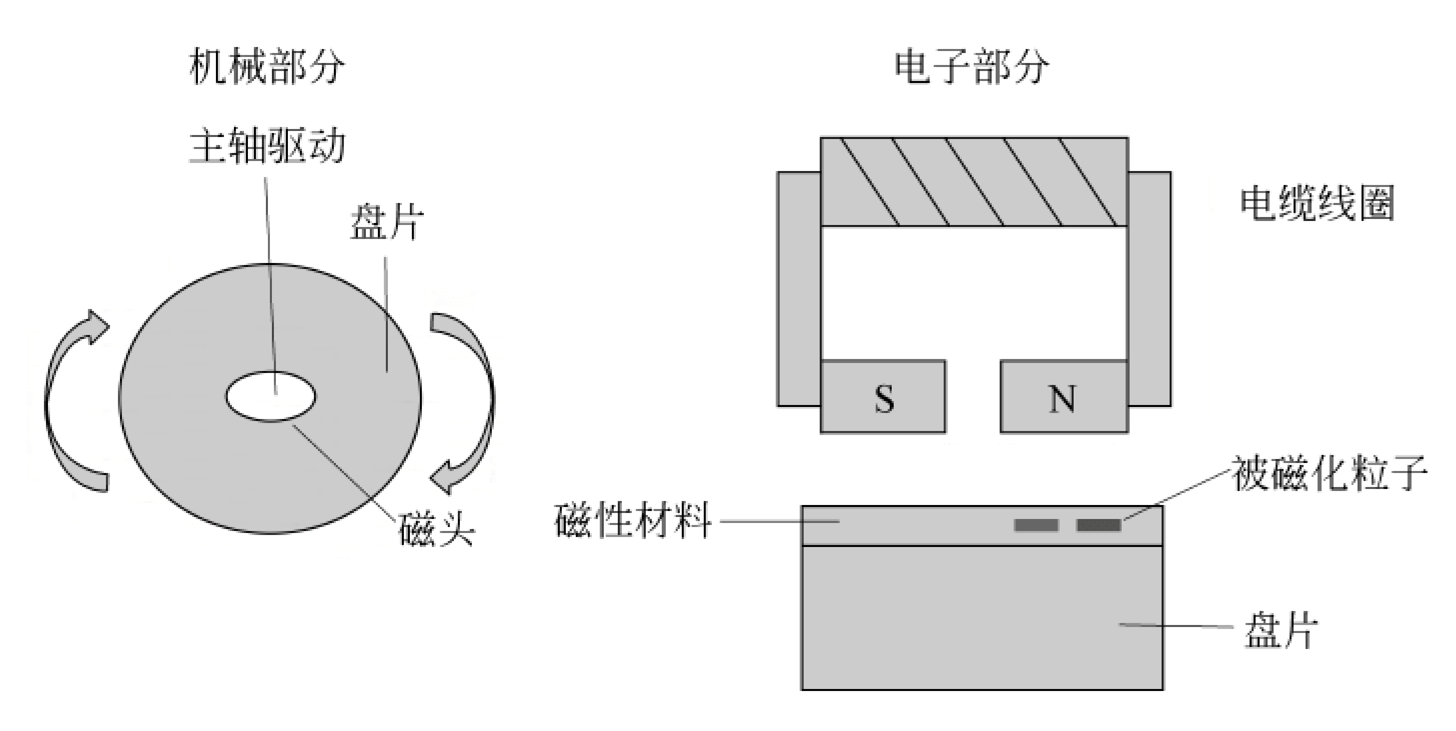


图 2-12　机械硬盘

(1) 机械部分

① 系统在密封机构里。

② 盘片固定并由主轴驱动来进行高速旋转。

③ 磁头沿盘片径向移动。

④ 磁头在盘片上方飞行。

(2) 电子部分

① 盘片上溅镀金属性粒子，呈不规则排列。

② 通过控制线圈上的电流，磁头形成磁场。

③ 对盘面上的金属粒子进行磁化(整齐排列)。

3. 盘片的功能分布

磁头靠近主轴接触的表面，即线速度最小的地方，是一个特殊的区域，它不存放任何数据，称为启停区或着陆区(Landing Zone)，启停区外就是数据区。在最外圈，离主轴最远的地方是0磁道，硬盘数据的存放就是从最外圈开始的。那么，磁头是如何找到0磁道的位置的呢？在硬盘中还有一个叫0磁道检测器的构件，它是用来完成硬盘的初始定位。

4. 盘片的逻辑结构

(1) 磁道(Track)：磁盘上人为规定的若干个同心圆的轨道称为磁道。磁道从外向内由0开始编号，数量为300～1024，甚至更多。

(2) 柱面(Cylinder)：所有盘面上的同一磁道构成一个圆柱，称为柱面；柱面从外向内由0开始编号，与磁道数目一致。

(3) 扇区(Sector)：磁盘上每个磁道被分为若干个弧段，从1开始编号。每个弧段可以存储512B或4KB的信息，称为扇区。

(4) 磁头数(Head Number)：磁盘上每个盘面都有对应的读写磁头。磁头数与盘面数一致，如图2-13所示。

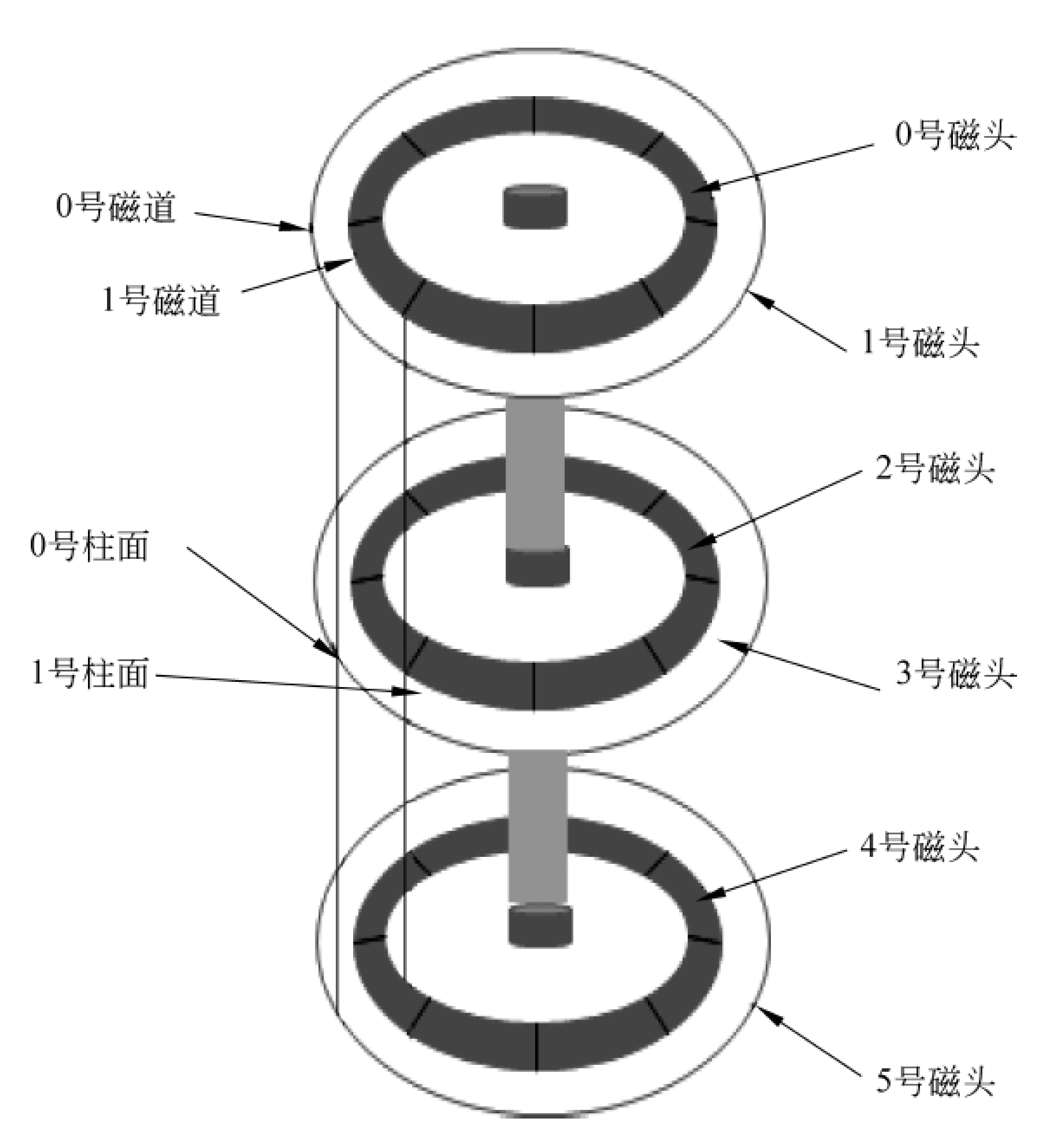


图2-13　盘片的逻辑结构

5. 硬盘主要参数

(1) 硬盘容量(Volume)：容量的单位为兆字节(MB)或千兆字节(GB)。影响硬盘容量的因素有单碟容量和碟片数量。

(2) 转速(Rotational Speed)：硬盘的转速是指硬盘盘片每分钟转过的圈数，单位为RPM(Rotation Per Minute)。一般硬盘的转速都达到5400/7200RPM。有些SCSI接口的硬盘使用了液态轴承技术，转速可达10000～15000RPM。

(3) 缓存(Cache)：由于CPU与硬盘之间存在巨大的速度差异，为解决硬盘在读写数

据时 CPU 的等待问题，在硬盘上设置了适当的高速缓存，以解决二者之间速度不匹配的问题。硬盘缓存与 CPU 上的高速缓存作用一样，是为了提高硬盘的读写速度。

6. 平均访问时间

(1) 平均寻道时间(Average Seek Time)：硬盘的平均寻道时间是指硬盘的磁头从初始位置移动到盘面指定磁道所需的时间，是影响硬盘内部数据传输率的重要参数。这个时间越小越好。目前 IDE 硬盘的平均寻道时间通常在 8～11ms。

(2) 硬盘的等待时间(Average Latency Time)：硬盘的等待时间又叫潜伏期，是指磁头已处于要访问的磁道，等待所要访问的扇区旋转至磁头下方的时间。平均等待时间通常为盘片旋转一周所需时间的一半，因此硬盘转速越快，等待时间就越短，一般应在 4ms 以下。

(3) 平均访问时间(Average Access Time)：平均访问时间＝平均寻道时间＋平均等待时间。

7. 数据传输率

(1) 数据传输率(Data Transfer Rate)：硬盘的数据传输率是指硬盘读写数据的速度，单位为兆字节每秒(MB/s)。硬盘数据传输率包括内部传输率和外部传输率两个指标。

(2) 内部传输率(Internal Transfer Rate)：内部传输率也称为持续传输率(Sustained Transfer Rate)，是缓存之间的数据传输速度，它反映了硬盘缓冲区没有使用时的性能，这可以说是影响硬盘整体速度的瓶颈。内部传输率主要依赖于硬盘的磁头到硬盘的高速旋转速度，并以 Mb/s 或 Mbps 为单位。

(3) 外部传输率(External Transfer Rate)：外部传输率也称为突发数据传输率(Burst Data Transfer Rate)或接口传输率。它指的是系统总线与硬盘缓冲区之间的数据传输率，与硬盘接口类型和硬盘缓存的大小有关。

8. IOPS 和 Throughput

(1) IOPS：IOPS(Input/Output Per Second)即每秒的输入输出量(或读写次数)，是衡量磁盘性能的主要指标之一。

(2) Throughput：Throughput(吞吐量)指单位时间内可以成功传输的数据数量。对于大量顺序读写的应用，如电视台的视频编辑、视频点播 VOD(Video On Demand)，则更关注吞吐量指标。

(3) IOPS 计算方法：磁盘完成一个 I/O 请求所花费的时间，它由寻道时间、旋转延迟和数据传输时间三部分构成。

寻道时间(Seek Time)是指将读写磁头移动至正确的磁道上所需要的时间。寻道时间越短，I/O 操作越快，目前磁盘的平均寻道时间一般为 3～15ms。

旋转延迟时间(Rotational Latency Time)是指盘片旋转将请求数据所在扇区移至读写磁头下方所需要的时间。旋转延迟取决于磁盘转速，通常使用磁盘旋转一周所需时间的 1/2 表示。比如，7200rpm 的磁盘平均旋转延迟大约为 60×1000/7200/2＝4.17ms，而转速为 15000 rpm 的磁盘其平均旋转延迟约为 2ms。

数据传输时间(Data Transfer Time)是指完成传输所请求的数据所需要的时间，它取决于数据传输率，其值等于数据大小除以数据传输率。目前 IDE/ATA 能达到 133MB/s，SATA Ⅱ可达到 300MB/s 的接口数据传输率，数据传输时间通常远小于前两部分时间。因此，理论上可以计算出磁盘的最大 IOPS，即 IOPS＝1000 ms/(寻道时间＋数据延迟时间)，

忽略数据传输时间。假设磁盘平均物理寻道时间为 3ms，磁盘转速分别为 7200rpm、1000rpm、15000rpm，则磁盘 IOPS 理论最大值分别为：

$$\text{IOPS} = 1000 \div (3 + 60000 \div 7200 \div 2) = 140$$

$$\text{IOPS} = 1000 \div (3 + 60000 \div 10000 \div 2) = 167$$

$$\text{IOPS} = 1000 \div (3 + 60000 \div 15000 \div 2) = 200$$

9. 硬盘常用接口

(1) ATA 接口

ATA 接口发展到今天，可以细分成 ATA-1(IDE)、ATA-2(EIDE Enhanced IDE/Fast ATA)、ATA-3(Fast ATA-2)、Ultra ATA、Ultra ATA/33、Ultra ATA/66、Ultra ATA/100 及 Serial ATA。ATA 发展到 ATA100 已经到了尽头，转向 SATA。

ATA 接口具有以下优缺点。

优点：价格低廉，兼容性非常好。

缺点：速度慢，只能内置使用，对接口电缆的长度有很严格的限制。

(2) SCSI 接口

SCSI(Small Computer System Interface，小型计算机系统接口)硬盘并发处理性能优异，常应用于企业级存储领域。SCSI 硬盘分 50 针、68 针、80 针，由 SCSI-1 不断发展至当前主流的 Ultra 320(320MB/s)，如图 2-14 所示。

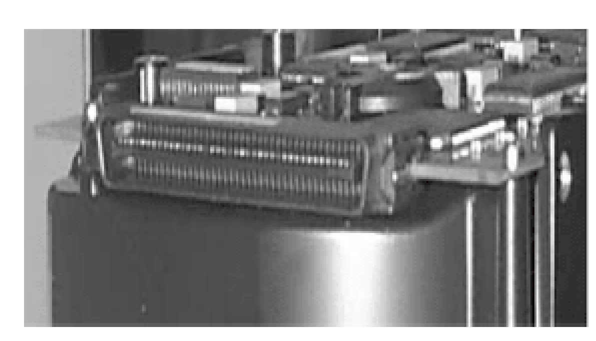



图 2-14　SCSI 接口

优点：

- 适应面广，在一块 SCSI 控制卡上就可以同时挂接 15 个设备。
- 高性能(具有多任务、高带宽及 CPU 占用率低等特点)。
- 有外置和内置两种，支持热插拔。

缺点：

价格昂贵，安装复杂。

(3) SATA 接口

SATA(Serial ATA，串行 ATA)采用串行方式进行数据传输，接口速率比 IDE 接口高，最低为 150MBps，并且第二代(SATA Ⅱ)300MBps 接口硬盘已经形成商用，规划内的最高速率可达 600MBps。

SATA 硬盘采用点对点连接方式，支持热插拔，即插即用。SATA 接口通常为 7+15 针，采用单通道，传输速率要比 ATA 更快。SATA 具有比 ATA 更好的抗干扰能力，如图 2-15 所示。

(4) SAS 接口

SAS(Serial Attached SCSI，串行连接 SCSI)是一种点对点、全双工、双端口的接口。其具有如下特点。



图 2-15　SATA 接口

① SAS 专为满足高性能企业需求而设计，实现与 SATA 的互操作，为企业用户带来前所未有的灵活性和低成本。

② 速率为每路 600Mbps。

SAS 具有高性能、高可靠性，以及强大的扩展能力。

SAS 可以向下兼容 SATA，同样采用串行技术，在传输速率、抗干扰性方面强于 SCSI，SAS 接口硬盘价格相对更高，如图 2-16 所示。

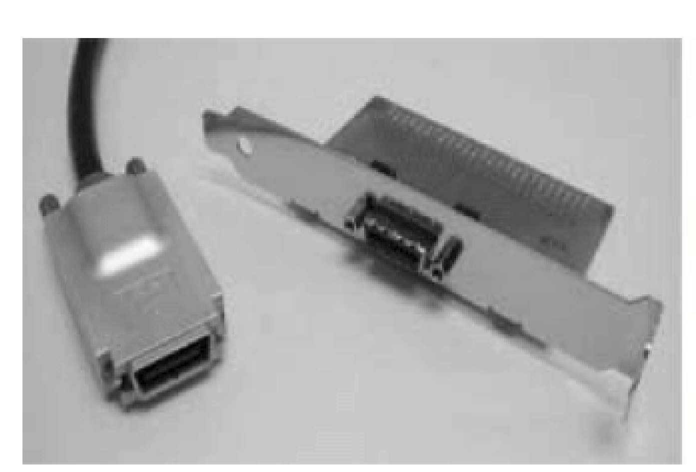




图 2-16　SAS 接口

(5) FC 接口

FC(光纤通道仲裁环)硬盘定位于高端存储应用，可靠性好，性能高。FC 硬盘一般同时提供两个 FC 接口，可同时使用或互为备份，如图 2-17 所示。

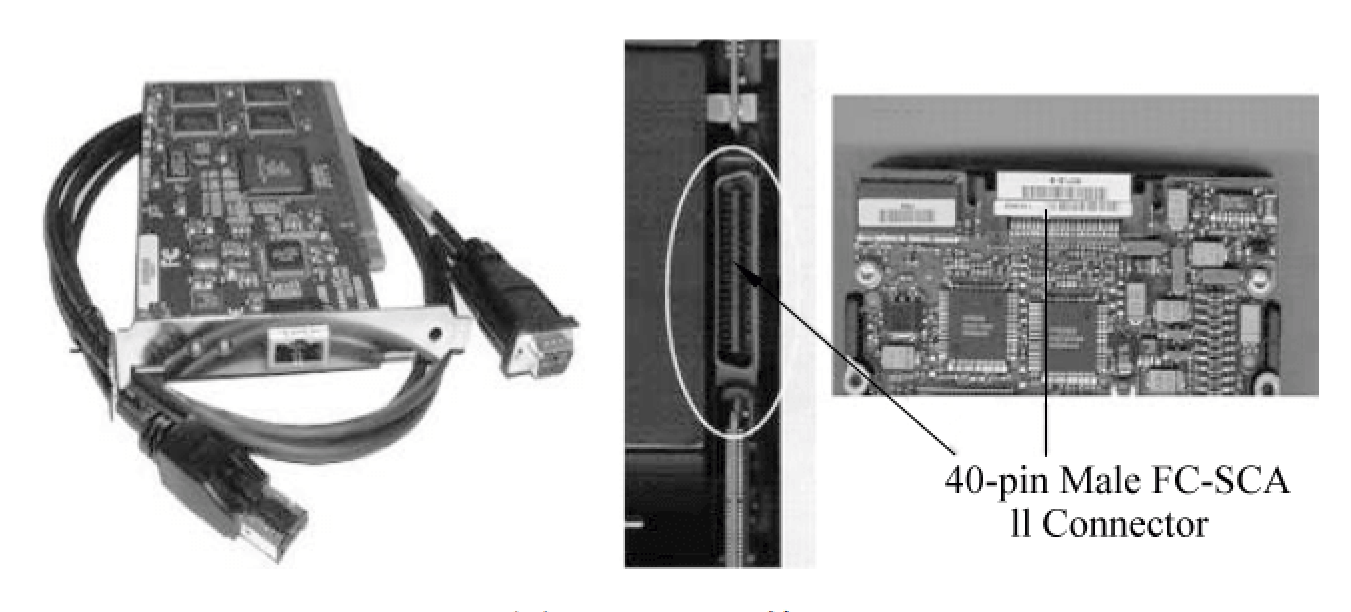


图 2-17　FC 接口

优点：

- 具有很好的升级性，可以用非常长的光纤电缆连接，可超过 10km；
- 具有非常高的带宽；
- 具有很强的通用性。

缺点：

- 价格非常昂贵；
- 组件复杂。

2.3.2 SSD 硬盘

1. SSD 简介

SSD 固态硬盘(Solid State Drives)简称固盘。固态硬盘是用固态电子存储芯片阵列而制成的硬盘，由控制单元和存储单元(FLASH 芯片、DRAM 芯片)组成。固态硬盘在接口的规范和定义、功能及使用方法上与普通硬盘完全相同，在产品外形和尺寸上也与普通硬盘完全一致，被广泛应用于军事、车载、工控、视频监控、网络监控、网络终端、电力、医疗、航空、导航设备等领域。

其芯片的工作温度范围很宽，商规产品为 0℃～70℃，工规产品－40℃～85℃。虽然成本较高，但也正在逐渐普及到 DIY 市场。由于固态硬盘技术与传统硬盘技术不同，所以产生了不少新兴的存储器厂商。厂商只需购买 NAND 存储器，再配合适当的控制芯片，就可以制造固态硬盘了。新一代的固态硬盘普遍采用 SATA-2 接口、SATA-3 接口、SAS 接口、MSATA 接口、PCI-E 接口、NGFF 接口、CFast 接口和 SFF-8639 接口，如图 2-18 所示。



图 2-18　SSD 固态硬盘

2. 分类方式

固态硬盘的存储介质分为两种，一种是采用闪存(FLASH 芯片)作为存储介质，另外一种是采用 DRAM 作为存储介质。

(1) 基于闪存类

基于闪存的固态硬盘(IDE FLASH DISK、Serial ATA Flash Disk)：采用 FLASH 芯片作为存储介质，这也是通常所说的 SSD。它的外观可以被制作成多种模样，例如，笔记本硬盘、微硬盘、存储卡、U 盘等样式。这种 SSD 固态硬盘最大的优点就是可以移动，而且数据保护不受电源控制，能适应于各种环境，适合于个人用户使用，如图 2-19 所示。

(2) 基于 DRAM 类

基于 DRAM 的固态硬盘采用 DRAM 作为存储介质，应用范围较窄。它效仿传统硬盘的设计，可被绝大部分操作系统的文件系统工具进行卷设置和管理，并提供工业标准的 PCI 和 FC 接口用于连接主机或者服务器。应用方式可分为 SSD 硬盘和 SSD 硬盘阵列两种。它是一种高性能的存储器，而且使用寿命很长，美中不足的是需要独立电源来保护数据安全。DRAM 固态硬盘属于比较非主流的设备，如图 2-20 所示。



图 2-19 基于闪存的固态硬盘



图 2-20 基于闪存的固态硬盘

3. 控制器

SSD 的关键部件由控制器和存储单元两部分组成。除此之外,还有缓存和主机接口。每个 SSD 都有一个控制器(controller)将存储单元连接到计算机上。控制器是一个执行固件(firmware)代码的嵌入式处理器。主要功能如下:

(1) 错误检查和纠正(ECC);

(2) 磨损平衡(Wear Leveling);

(3) 坏块映射(Bad Block Mapping);

(4) Read Disturb(读取某个块的数据的时候会影响到相邻块的数据)管理;

(5) 缓存控制;

(6) 垃圾回收;

(7) 加密。

4. 存储单元

尽管某些厂商推出了基于更高速的 DRAM 内存的产品,但 NAND 闪存依然最常见,占据着绝对主导地位。低端产品一般采用 MLC(Multi Level Cell)甚至 TLC(Triple Level Cell)闪存,其特点是容量大、速度慢、可靠性低、存取次数低、价格也低。高端产品一般采用 SLC(Single Level Cell)闪存,其特点是技术成熟、容量小、速度快、可靠性高、存取次数高、价格也高。但是事实上,取决于不同产品的内部架构设计,速度和可靠性的差别也可以通过各种技术加以弥补甚至反转。

5. 缓存

基于 NAND 闪存的 SSD 通常带有一个基于 DRAM 的缓存,其作用与普通的机械式硬盘类似,但是还会存储一些诸如 Wear Leveling 数据之类的其他数据。把数据先缓存在 DRAM 中,然后集中写入,从而减少写入次数。特例之一是 SandForce 公司生产的控制器,它不含缓存,但是性能依旧很出色,由于其结构简单,故而可以生产体积更小的 SSD,并且掉电时数据更安全。

6. 主机接口

主机接口与控制器紧密相关,但是通常与传统的机械式硬盘相差不大,主要有①SATA;②SAS;③PCI-E;④Fibre Channel;⑤USB。主机接口的速度限制了 SSD 所能达到的速度峰值,但是一般这并不是导致瓶颈的原因。

7. SSD 性能

SSD 性能大致上可以用并行工作的 NAND 闪存芯片数(也称为通道数,目前主流的固态硬盘拥有 8~16 通道)来衡量。一个单独的 NAND 芯片很慢,但是当多个芯片

并行工作时，性能就会得到巨大的提升，其原理类似于RAID0，买块SSD就等于是买了块“阵列卡+盘”。

在2010年由著名的Xssist网站使用IOmeter工具对Intel X25-E 64 GB G1进行的一项测试中，以4KB随机“70%读+30%写”、队列深度为4，进行持续测试。开始的时候可以达到10 000 IOPS，但是性能很快便急速下降，8分钟之后就只剩下4 000 IOPS了，而50分钟之后，就稳定在3 500 IOPS左右了。除了这种短时间可发现的急速性能下降之外，还存在着随时间推移性能缓慢下降的问题(主要是存储单元老化和损坏所致)。能否有效地处理性能下降的问题不但关系到SSD的长期运行的实际性能，而且关系到其寿命(SSD一旦损坏，其内部的数据将全部丢失，而且基本上是不可恢复的)。因为导致性能下降的原因也和其使用寿命紧密相关。通常，固态硬盘的性能越差意味着它的使用寿命就越短。这是因为固态硬盘的磨损与固态硬盘上发生的数据写入和清除次数直接相关。发生写数据的次数越多，性能就越差，其寿命也就越短。

对于传统硬盘，人们常用平均无故障时间(MBTF)来标识其可靠性，现在很多SSD制造商借用这个指标来说明SSD质量的高低。显然，这样做法过于牵强。事实上，SSD的寿命与其如何使用有密切关系。比如，Intel的消费级SSD X25-M的MTBF为120万个小时，与普通的磁介质大体相当。Intel估计，如果每天写入100GB数据，理论上可以使用5年，不过这只是理论上的最优情况，实际寿命肯定比这个要短。另外，NAND闪存中的数据最多只可以保存10年左右，也就是说10年是SSD理论上的最大寿命。值得注意的是，SSD的使用寿命主要取决于其写入数据的次数，而与读取次数关系不大。有鉴于此，那些以“一次存取，多次查询”为主的应用形式(如搜索引擎、数据仓库)应该是SSD最适合的应用场合。

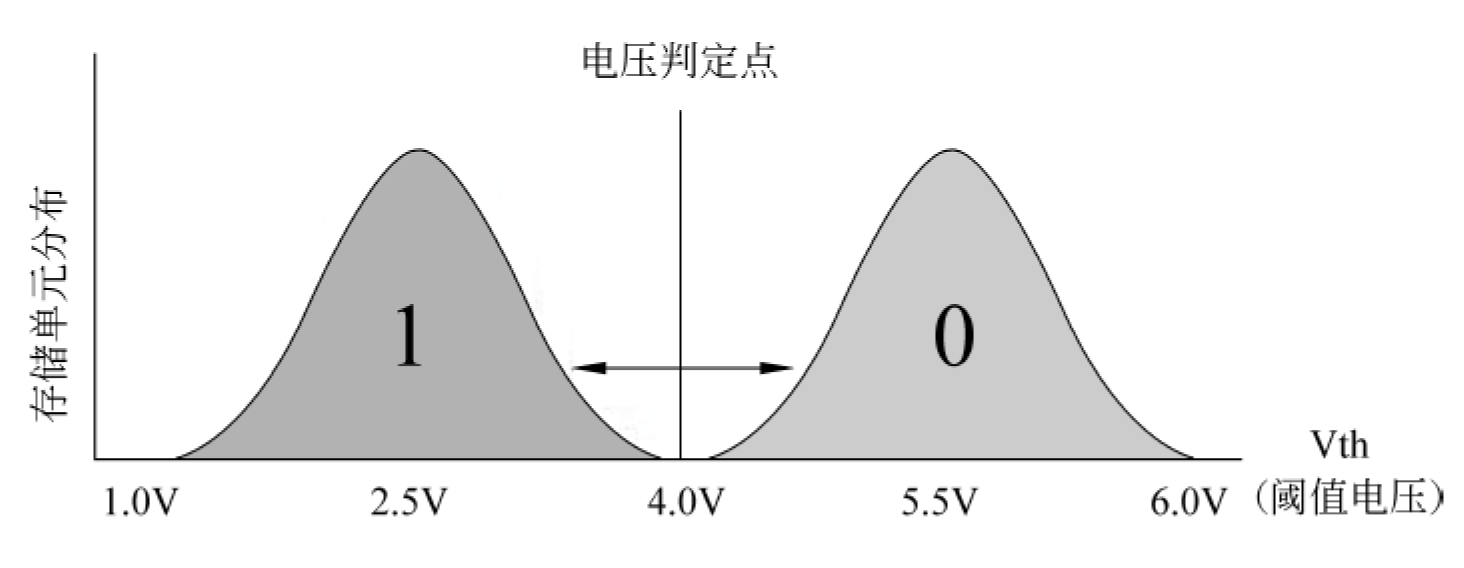
8. SSD技术解析

(1) SLC/eSLC、MLC/eMLC和TLC/eTLC

SSD的基本存储单元分为三类：SLC(Single Level Cell，单层单元)和MLC(Multi Level Cell，多层单元)以及较新的TLC(Triple Level Cell，三层单元)。低端产品一般采用MLC(multi Level Cell)或者TLC(Triple Level Cell)闪存，其特点是功耗高、容量大、速度慢(2MB/s)、可靠性低、存取次数低(3000次(25nm)～10 000次(50nm)，制程越先进次数反而越小)、价格也低。高端产品一般采用SLC(Single Level Cell)闪存，其特点是技术成熟、功耗低、容量小、速度快(8MB/s)、可靠性高、存取次数高(10万次)、价格也高。造成这种差异的原因在于，每个MLC/TLC存储单元中存放的资料较多，结构相对复杂，出错的概率会增加，必须进行错误修正，这个动作导致其性能和可靠性大幅落后于结构简单的SLC闪存。

SLC就是在每个存储单元里存储1bit的数据，存储的数据是0还是1基于电压阈值的判定，对于NAND Flash的写入(编程)，就是控制Control Gate去充电(对Control Gate加压)，使得浮置栅极存储的电荷够多，超过4V，存储单元就表示0(已编程)；如果没有充电或者电压阈值低于4V，就表示1(已擦除)，如图2-21所示。

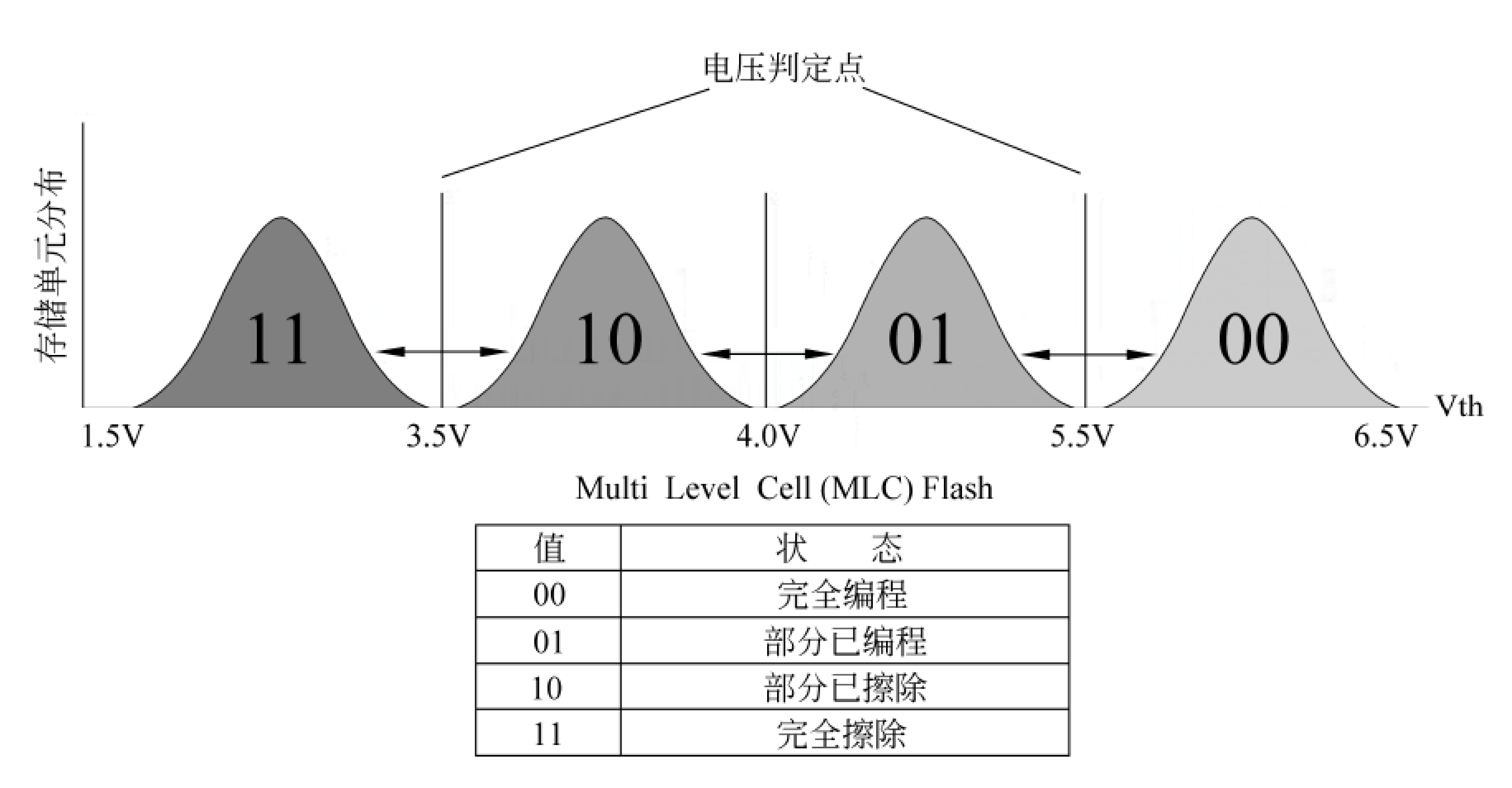
MLC就是每个存储单元里存储2bit的数据，存储的数据是00、01、10、11也是基于电压阈值的判定，当充入的电荷不足3.5V时，就代表11；当充入的电荷在3.5V和4.0V之间，则代表10；当充入的电荷在4V和5.5V之间，则表示01；当充入的电荷在5.5V以上，则表示00。同时MLC相比SLC虽然使用相同的电压值，但是电压之间的阈值被分成了4份，这样就直接影响了性能和稳定性，如图2-22所示。



Single Level Cell (SLC) Flash

值	状态
0	已编程
1	已擦除

图 2-21　SSD 的基本存储单元 SLC 单层单元



值	状　态
00	完全编程
01	部分已编程
10	部分已擦除
11	完全擦除

图 2-22　SSD 的基本存储单元 MLC 单层单元

而 TLC 更加复杂，因为每个存储单元里存储 3bit 的数据，所以它的电压阈值的分界点就更细致，导致的结果也就是每个存储单元的可靠性更低。由于 TLC 与 MLC 没有本质区别，所以在本文剩余部分就不再提及 TLC 了。

在 NAND Flash 工厂制造处理过程中，厂商把晶元上最好的那部分 Flash 晶片挑选出来并用企业级的标准来检测晶片的数据完整性和耐久度。检测完后，这些晶片被取下来改变内部些许参数并进行之后的比标准 SLC/MLC 更苛刻的测试。当这些晶片通过测试后，就被定义为 eSLC/eMLC 级别组，余下的就成为普通 SLC/MLC 级别组了。

相对普通 SLC/MLC 来说，eSLC/eMLC 的不同之处主要体现在下面 4 个方面。

① P/E 数字更大，比如 34nm 镁光的 eMLC 是 30 000 次，而 MLC 是 5 000 次。

② eSLC/eMLC 擦写操作和编程操作所需要的时间相比 SLC/MLC 更长。

③ 当使用完厂商保证的 P/E 数后，eSLC/eMLC 的数据保存期一般在 3 个月，而 SLC/MLC 的数据保存期在 1 年。

④ 相对于企业级的应用下，使用 eSLC/eMLC 的稳定性比 SLC/MLC 要高得多，也就

是出错的概率更小。

（2）多 Plane NAND

多 Plane NAND 是一种能够有效提升性能的设计。多 Plane 的原理很简单，我们看到，一个晶片内部（Micron 25nm L73A）分成了 2 个 Plane，而且 2 个 Plane 内的 Block 编号是单双交叉的。我们在操作时，也可以进行交叉操作（一单一双）来提升性能。根据测试，某些情况下性能可以比单 Plane 设计提高约 50％以上，如图 2-23 所示。

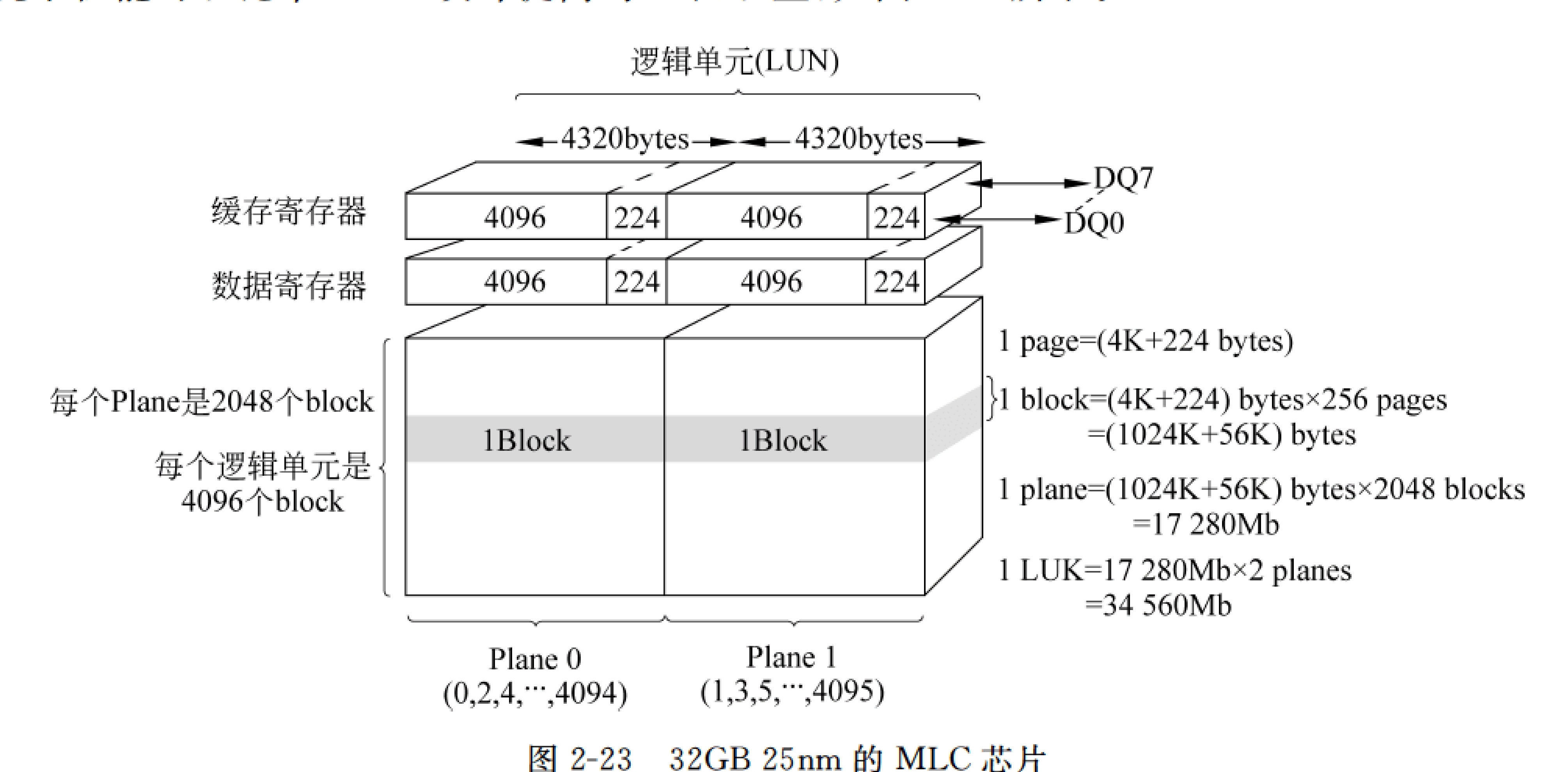


图 2-23　32GB 25nm 的 MLC 芯片

我们看到两个 Plane Page 读取操作相比单个页面（Page）读取操作，每两个页面节省了一次页面的读取时间。同样作为擦除、写入操作，两个 Plane 的交叉操作也能带来性能的提升，如图 2-24 所示。

（3）磨损平衡

磨损平衡（Wear Leveling）是确保闪存的每个块被写入的次数相等的一种机制。通常情况下，在 NAND 块里的数据更新频度是不同的：有些会经常更新，有些则不常更新。很明显，那些经常更新的数据所占用的块会被快速地磨损掉，而不常更新的数据占用的块磨损就小得多。为了解决这个问题，需要让每个块的编程（擦写）次数尽可能保持一致：这就需要对每个页的读取/编程操作进行监测，在最乐观的情况下，这个技术会让全盘的颗粒物理磨损程度相同并同时报废。

磨损平衡算法分静态和动态。动态磨损算法是基本的磨损算法：只有用户在使用中更新的文件占用的物理页地址被磨损平衡了。而静态磨损算法是更高级的磨损算法：在动态磨损算法的基础上，增加了对于那些不常更新的文件占用的物理地址进行磨损平衡，这才算是真正的全盘磨损平衡。简单说来，动态算法就是每次都挑最新的 NAND 块来用，老的 NAND 块尽量不用。静态算法就是把长期没有修改的老数据从一个新 NAND 块里面搬出来，重新找个最老的 NAND 块放着，这样新的 NAND 块就能再度进入经常使用区。概念很简单，但实现却非常复杂，特别是静态算法。

尽管磨损均衡的目的是避免数据重复在某个空间写入，以保证各个存储区域内磨损程度基本一致，从而达到延长固态硬盘的目的，但是，它对固态硬盘的性能有不利影响。

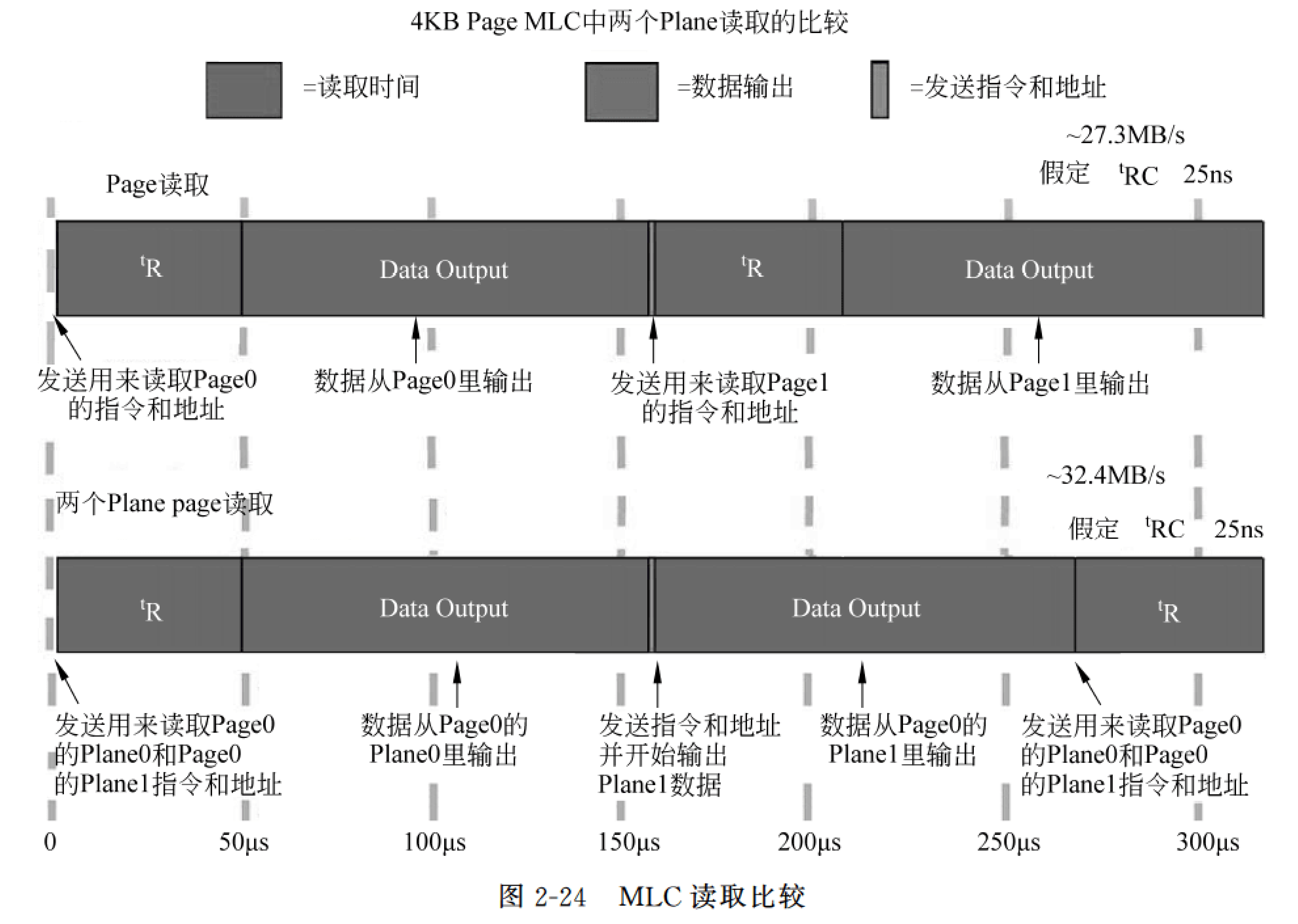


图 2-24　MLC 读取比较

（4）垃圾回收

由前面的磨损平衡机制知道，磨损平衡的执行需要有“空白块”来写入更新后的数据。当可以直接写入数据的“备用空白块”数量低于一个阈值后，SSD 主控制器就会把那些包含无效数据的块里的所有有效数据合并起来写到新的“空白块”中，然后擦除这个块以增加“备用空白块”的数量。这个操作就是 SSD 的垃圾回收（Garbage Collection）。有三种垃圾回收策略，如图 2-25 所示。

① 闲置垃圾回收。很明显在进行垃圾回收时会消耗大量的主控处理能力和带宽造成处理用户请求的性能下降，SSD 主控制器可以设置在系统闲置时做“预先”垃圾回收（提前做垃圾回收操作），保证一定数量的“备用空白块”，让 SSD 在运行时能够保持较高的性能。闲置垃圾回收的缺点是会增加额外的“写入放大”，因为用户刚刚垃圾回收的“有效数据”，也许马上就会被更新后的数据替代而变成“无效数据”，这样就造成之前的垃圾回收做无用功了。

② 被动垃圾回收。这是每个 SSD 都支持的技术，但是对主控制器的性能提出了很高的要求，适合在服务器里用到，SandForce 的主控就属于这类。在垃圾回收操作消耗带宽和处理能力的同时处理用户操作数据，如果没有足够强劲的主控制器性能则会造成明显的速度下降。这就是为啥很多 SSD 在全盘写满一次后会出现性能下降的原因，因为要想继续写入数据，就必须要边回收垃圾边做写入。

③ 手动垃圾回收。用户自己手动选择合适的时机运行垃圾回收软件，执行垃圾回收操作。可以想象，如果系统经常进行垃圾回收处理，频繁地将一些区块进行擦除操作，那么 SSD 的寿命也会进一步下降。由此把握这个垃圾回收的频繁程度，同时确保 SSD 中的闪存



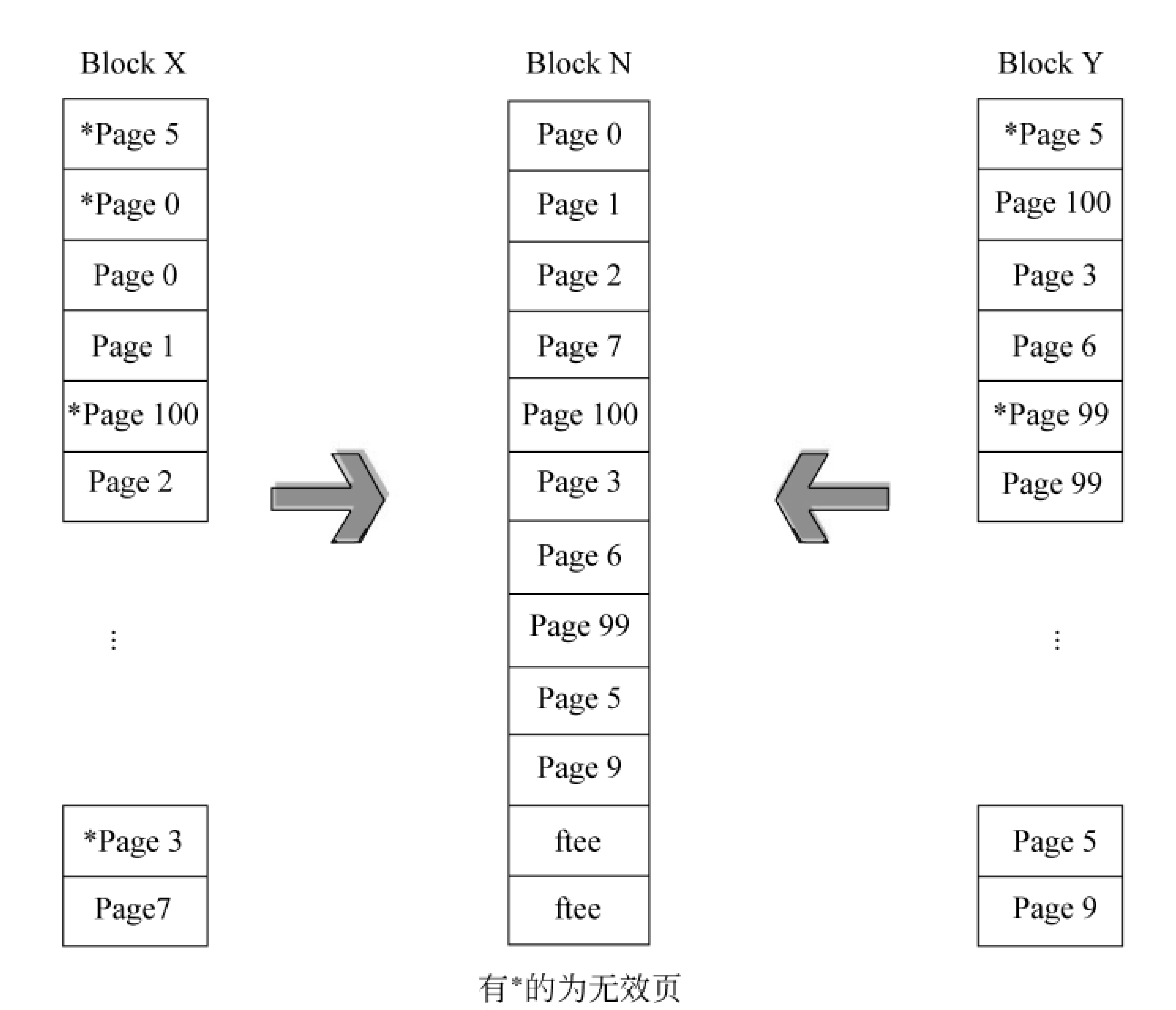


图 2-25　基本垃圾回收(GC)原理图

芯片拥有更高的使用寿命。

(5) 预留空间

预留空间(Over Provisioning,OP)是指用户不可操作的容量,为实际物理闪存容量减去用户可用容量。这块区域一般被用来做优化,包括磨损均衡,GC 和坏块映射,如图 2-26 所示。

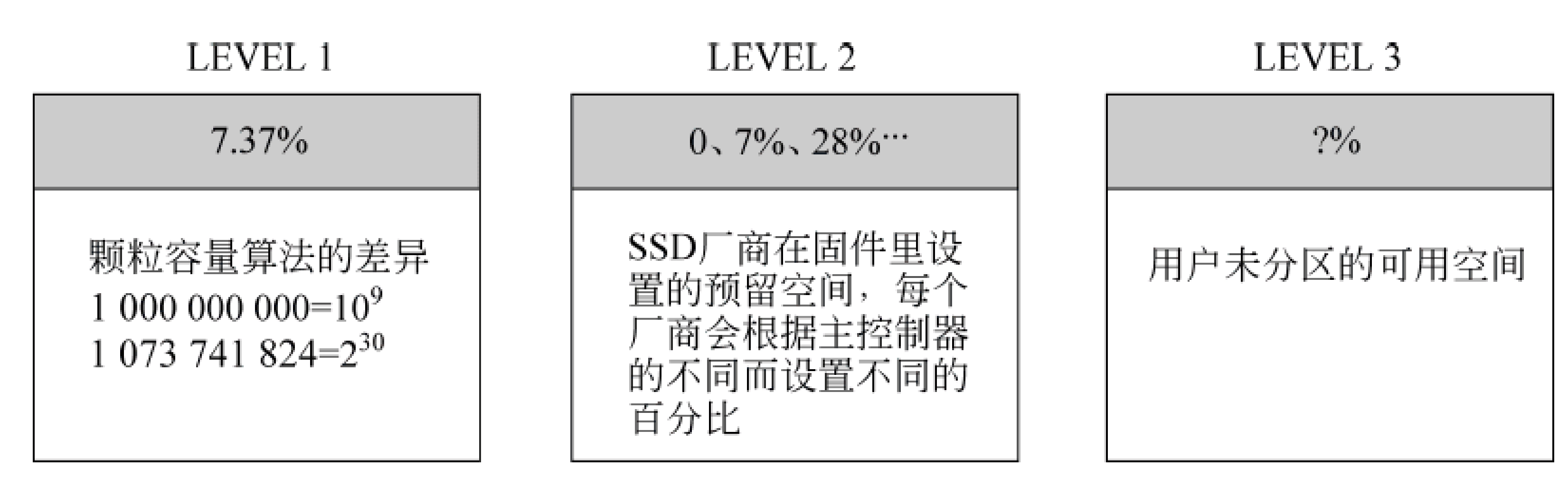


图 2-26　SSD 预留空间的三层

第一层为固定的 7.37%,这个数字是如何得出的呢? 我们知道,机械硬盘和 SSD 的厂商容量是这样算的,1GB 是 1 000 000 000 字节(10^9),但是闪存的实际容量是每 GB=1 073 741 824,(2^{30}),两者相差 7.37%。所以说假设 1 块 128GB 的 SSD 用户得到的容量是 128 000 000 000 字节,多出来的那个 7.37%就被主控固件用做预留空间了。

第二层来自制造商的设置,通常为 0、7%、28%等,打个比方,对于 128GB 颗粒的 SandForce 主控 SSD,市场上会有 120GB 和 100GB 两种型号卖,这取决于厂商的固件设置,这个容量不包括之前第一层的 7.37%。

第三层是用户在日常使用中可以分配的预留空间,用户可以在分区的时候,不分到完全的 SSD 容量来达到这个目的。不过需要注意的是,需要先做安全擦除(Secure Erase),以保证此空间确实没有被使用过。

预留空间虽然让 SSD 的可用容量小了,但是带来了减少写入放大、提高耐久性、提高性能的效果。根据经验,预留空间在 20%~35%是最佳平衡点。

(6) 写入放大

因为闪存必须先擦除(也叫编程)才能写入,在执行这些操作的时候,移动或覆盖用户数据和元数据(metadata)不止一次。这些额外的操作,不但增加了写入数据量,减少了 SSD 的使用寿命,而且还用光了闪存的带宽,间接地影响了随机写入性能。这种效应就叫写入放大(Write Amplification)。一个主控的好坏主要体现在写入放大上。

比如要写入一个 4KB 的数据,最坏的情况是,一个块里已经没有干净空间了,但是有无效数据可以擦除,所以主控就把所有的数据读到缓存并擦除块,从缓存里更新整个块的数据,再把新数据写回去。这个操作带来的写入放大就是:实际写 4KB 的数据,却造成了整个块(1024KB)的写入操作,那就是 256 倍放大。同时带来了原本只需要简单的写 4KB 的操作变成闪存读取(1024KB)、缓存改(4KB)、闪存擦(1024KB)、闪存写(1024KB),造成了延迟的大大增加,速度急剧下降也就是很自然的事情了。所以,写入放大是影响 SSD 随机写入性能和寿命的关键因素。

用 100%随机 4KB 来写入 SSD,对于目前的大多数 SSD 主控而言,在最糟糕的情况下,写入放大的实际值可能会达到或超过 20 倍。当然,用户也可以设置一定的预留空间来减少

写入放大，假设有个128GB的SSD，则只分了64GB的区使用，那么最坏情况下的写入放大就能减少约3倍。

许多因素影响SSD的写入放大。下面列出了主要因素，以及它们如何影响写入放大。

① 垃圾回收虽然增加了写入放大（被动垃圾回收不影响，闲置垃圾回收影响），但是速度有所提升。

② 预留空间可以减少写入放大，预留空间越大，写入放大越低。

③ 开启TRIM指令后可以减少写入放大。

④ 用户使用中没有用到的空间越大，写入放大越低（需要有Trim支持）。

⑤ 持续写入可以减少写入放大。理论上来说，持续写入的写入放大为1，但是某些因素还是会影响这个数值。

⑥ 随机写入将会大大提升写入放大，因为会写入很多非连续的LBA。

⑦ 磨损平衡机制直接提高了写入放大。

(7) DuraWrite

NAND闪存存储的一大缺陷就是需要在写入时对存储结构进行整理，这导致实际上写入的数据比我们真正需要存储的数据量大。在一款比较普通的固态硬盘中，如果需要写入1GB数据，在盘内结构已经比较混乱（存储、删除、再存储）的情况下，最后真正写入的数据量可能高达10GB甚至20GB。真实写入数据与需要写入数据之比即为“写入放大率”。

DuraWrite是SandForce开发的一种减少SSD写入放大率的技术：写入1GB数据时，最终写进闪存的可能只有500MB甚至更少。根据厂方的测试，安装Windows Vista和Office 2007的全过程共需要写入25GB数据，而使用DuraWrite技术实际写入仅为11GB。所以厂商自称其写入放大率是0.5，奥秘在于SandForce在将数据写入闪存前进行了压缩。不过根据厂方的说法，这并不仅仅是压缩这么简单，而是一套多种多样的数据缩量算法。比如当数据存在重复时，仅写入特殊部分；当数据可压缩时，即进行压缩再存储等。由于写入数据变少，SandForce控制器不需要使用外部DRAM缓存，而是在芯片内直接集成了较大容量的缓存。

这项技术确实可以带来很多优点，特别是性能上。写入的数据少了，相对来说速度自然就翻倍了，而读取操作同样如此。因此，数据库等类似的需要高吞吐量的操作都可以获得极佳的性能发挥。当然，SandForce标称高达500MB/s的写入速度只是在最好情况下的成绩（数据可以被实时压缩）而已，不可迷信。但这样的技术也有弱点，当需要写入的数据已经进行过压缩时（如图片、视频或压缩文件），其算法就无法再发挥理想效果。

其实DuraWrite是包含于DuraClass技术之中的一个组件。DuraClass技术包含RAISE、DuraWrite、GC、ECC等技术。RAISE是一项类似于RAID5的功能概制，是一个独立的冗余数组结构，这个功用的主要目标在于改进故障概率，保障压缩数据的安全。其实这项技术也是配合DuraWrite技术而运作的。所以可以说DuraWrite是DuraClass技术的关键，也是SandForce系列主控的灵魂。

(8) 坏块管理

不管磨损平衡算法如何优越，在运作中都会碰到一个令人头痛的问题，那就是坏块，所以一个SSD必须要有坏块管理机制。何谓坏块？即一个NAND块里包含有不稳定的地址，不能保证读/写/擦的时候数据的准确性。其概念和传统机械式硬盘的坏块相似，如

图 2-27 所示。

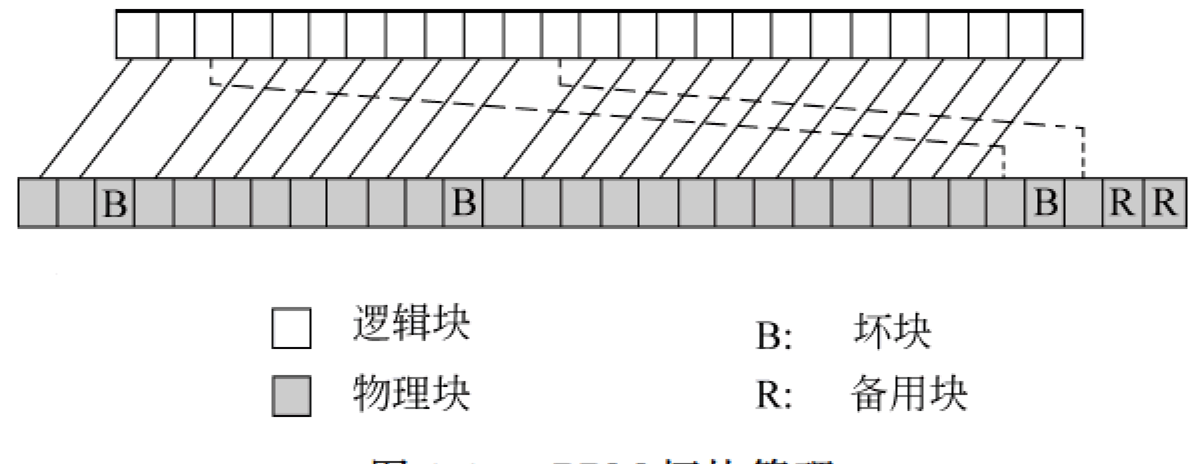


图 2-27　BBM 坏块管理

坏块分出厂坏块和使用过程中出现的坏块,与机械式硬盘的坏块表一样(P 表和G 表),SSD 也有坏块表。出厂如果有坏块,在坏块上会有标记,所以很容易就能被识别,后期使用中出现的坏块就要靠主控制器的能力了。一般来说,越到 NAND 生命的后期(P/E 数开始接近理论最大值),坏块就会开始大量出现了。NAND 出厂前都会被执行擦除操作,厂商会在出货前把坏块标记出来(厂商会在坏块的第一个页的 SA 区上打上标记)。这样坏块管理软件就能靠监测 SA 区标记来制作坏块表。SA 区的意思是页中 4096～4319 的区域,用户不可访问,主要用来存放 ECC 算法、坏块信息、文件系统资料等。由于在使用中会产生坏块,所以 SSD 的每次编程/擦除/复制等操作后都要检查块的状态。对颗粒的 ECC 要求也要达到厂商的标准以上(主控强不强,看 ECC 能力也是一个参考)。

(9) ECC

ECC 的全称是 Error Checking and Correction,是一种用于 Nand 的差错检测和修正算法。由于 NAND Flash 的工艺不能保证 NAND 在其生命周期中保持性能的可靠,因此,在 NAND 的生产中及使用过程中会产生坏块。为了检测数据的可靠性,在应用 NAND Flash 的系统中一般都会采用一定的坏区管理机制,而管理坏区的前提是能比较可靠地进行坏区检测。如果操作时序和电路稳定性不存在问题,NAND Flash 出错的时候一般不会造成整个 Block 或者 Page 不能读取或者全部出错,而是整个 Page 中只有一个或几个 bit 出错,这时候 ECC 就能发挥作用了。不同颗粒有不同的基本 ECC 要求,不同主控制器支持的 ECC 能力也不同,理论上说主控越强 ECC 能力越强。

(10) 交错操作

交错操作可以成倍提升 NAND 的传输率,因为 NAND 颗粒封装时可能有多 Die、多 Plane(每个 plane 都有 4KB 寄存器),Plane 操作时可以交叉操作(第一个 Plane 接到指令后,在操作的同时,第二个指令已经发送给了第二个 plane,以此类推),达到接近 2 倍甚至 4 倍的传输能力(看闪存颗粒支持度),如图 2-28 所示。

(11) 断电保护

这是英特尔在所有第三代固态硬盘中重点增加的一项功能,而在 SandForce 的方案中作为可选项提供。SandForce 的方案是配备了一个 0.09F 的大容量电容以保证断电之后数据还可以写入闪存当中,防止丢失。更进一步讲,还可以多电容并联带来更好的可靠性,就算其中一个坏掉,其他的电容还可以正常工作。

(12) SMART 监控

SMAR T(Self-Monitoring Analysis and Reporting Technology)中文的意思是“自我监

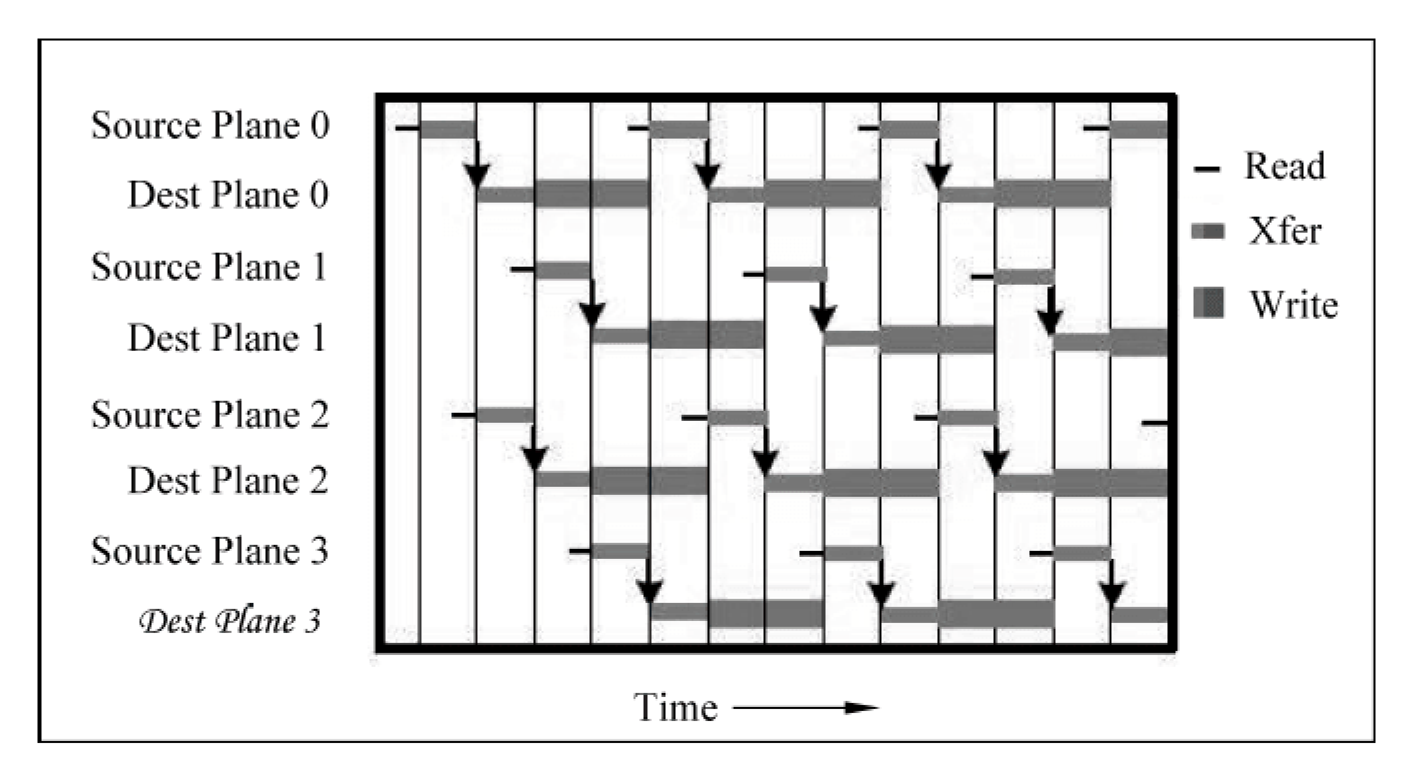


图 2-28　SSD 交错操作

测、分析和报告技术”,可以用来预测并分析硬盘的潜在问题。这是一项延续自传统机械式硬盘的技术。

硬盘故障分为忽然性和渐进性两种。

① 忽然性：没有任何迹象,很难预防。比如芯片烧了、硬盘摔了、人品不好之类。要防止只有多做备份,或者做 RAID 之类的冗余。

② 渐进性：这是随着时间慢慢发生的,可以预先感知。比如声音出现异常,可能是主轴马达磨损,硬盘逐渐老化出现读取困难等,在 SSD 上主要指颗粒磨损度,不可修复错误数明显增加等。

对于渐进性的故障,我们可以借助 SMART 数据发现点有用的信息。但是需要注意的是：机械硬盘的 SMART 表定义已经有自己的标准,由于硬盘厂很多,很多厂家属性的名字也不尽相同,或者某些厂牌缺少某些属性,但是同个 ID 的定义是相同的。而固态硬盘的 SMART 表定义则目前还没有统一标准,不同厂家甚至不同主控都有可能出现相同 ID 的不同定义,所以用一般的 SMART 软件查看是没任何意义的,虽然可以看到值,但是这个值对应的 ID 解释可能完全不是那么回事。

2.3.3　SSD 与 HDD 的比较

全面将 SSD 和 HDD 进行对比是很复杂的,传统的 HDD 性能评测主要关注其固有的弱点,比如寻道时间和转速。SSD 并不旋转,也不存在寻道的问题,故而在这些传统测试中,可以取得惊人的成绩。但是,SSD 有其自身特有的弱点,比如混合读写、垃圾回收、ECC、磨损平衡,等等。而且通常一个新 SSD 空盘的性能会比使用了一年之后、包含很多数据的 SSD 性能高出许多。所以面向传统 HDD 的性能测试方法并不适用于 SSD。

尽管难以简单对比,表 2-1 还是在理论上给出两者的差异。

表 2-1　SSD 与 HDD 各项参数比较

各项参数	SSD	HDD
启动时间	由于没有马达和转臂,所以几乎可以瞬间完成。同时从休眠模式中唤醒大约只需要几毫秒即可	可能需要数秒以启动马达,而且当磁盘量非常大的时候,需要依次启动以防止瞬间电流过载

续表

各项参数	SSD	HDD
随机访问时间	大约仅需 0.1 毫秒，因为无须寻道	大约需要 5～10 毫秒
读取潜伏期	通常很短，因为直接读取	通常比较高，因为磁头需要额外的时间等待扇区的到来
读取性能一致性	读取性能不因数据在 SSD 上的存储位置不同而不同	读取性能与存放在磁盘的内圈还是外圈有关，也与文件的碎片程度有关
碎片整理	SSD 基本不需要进行碎片整理，因为读取连续的数据并不明显比读取分散的数据快。并且碎片整理会额外增加 NAND 闪存的写入次数，从而降低其寿命	HDD 通常需要在文件碎片达到一定程度后进行整理，否则性能会有明显下降。特别是在含有大量文件的情况下更是如此
噪声	SSD 无任何噪声	HDD 有明显的噪声，并且在读写频繁的时候噪声更大
机械可靠性	无机械故障	随着时间的推移，机械故障概率会逐渐增加
环境敏感性	对振动、磁场、碰撞不敏感	对振动、磁场、碰撞敏感
体积和重量	体积小、重量轻	性能越高，体积和重量越大
并行操作	多数控制器可以使用多个芯片进行并发读写	HDD 虽然有多个磁头，但是由于共享同一个位置控制电机，所以不能并发读写
写入寿命	基于闪存的 SSD 有写入寿命限制，且一旦损坏，整个 SSD 的数据都将丢失	无写入寿命限制
数据安全问题	NAND 闪存的存储块不能被直接覆盖重写，只能重新写入先前被擦除的块中。如果一个软件加密程序对已经存在于 SSD 上的数据进行加密，那些原始的、看上去已经被覆盖掉的原始数据实际上并没有被覆盖，它们依然可以被读取，从而造成信息泄露。但是 SSD 自身基于硬件的加密装置没有这个问题。此外，也不能简单地通过覆盖原文件的办法来清除原有的数据，除非该 SSD 有内建的安全删除机制，并且确实已经被启用	HDD 可以直接覆盖掉指定的扇区，因而不存在这个问题
单位容量成本	贵，但是大约每两年下降一半	便宜
最大存储容量	小，但是大约每两年可翻一倍	大
读/写性能对称	低端 SSD 的读取速度远高于写入速度，但是高端产品的读写速度可以做到一致	HDD 的读取速度通常比写入速度快一些，但是差距并不很大
TRIM 与可用空白块	SSD 的写入性能受可用空白块数量影响很大。先前曾经写入过数据且现在未被使用的块，可以通过 TRIM 来回收，使其成为可用的空白块。但是即使经过 TRIM 回收的块，其性能依然会出现下降	HDD 完全没有这些问题，其性能不会因为多次读写而出现下降，也不需要进行 TRIM 操作
能耗	即使是高性能的 SSD 通常其能耗也只有 HDD 的 1/2～1/3	高性能 HDD 通常需要 12～18W，而为笔记本设计的节能 HDD 的功耗通常在 2～3W

NCQ(Native Command Queuing)与TCQ(Tagged Command Queuing)都是设计为通过把计算机发往硬盘的指令做重新排序从而提高硬盘性能的技术。NCQ技术在300MB/s的Serial ATA Ⅱ规格中引入,针对的是主流的硬盘产品,而TCQ技术是在SCSI2规格中引入(ATA-4标准中也有采用),针对的是服务器以及企业级硬盘产品。

要使用NCQ、TCQ技术,芯片组硬盘接口和硬盘产品本身都必须支持才行,也就是说,如果你购买的一款新硬盘并不支持NCQ,即使主板是最新的支持NCQ的,也不能够使用这个功能从而提高性能。

2.3.4 SSD性能优势

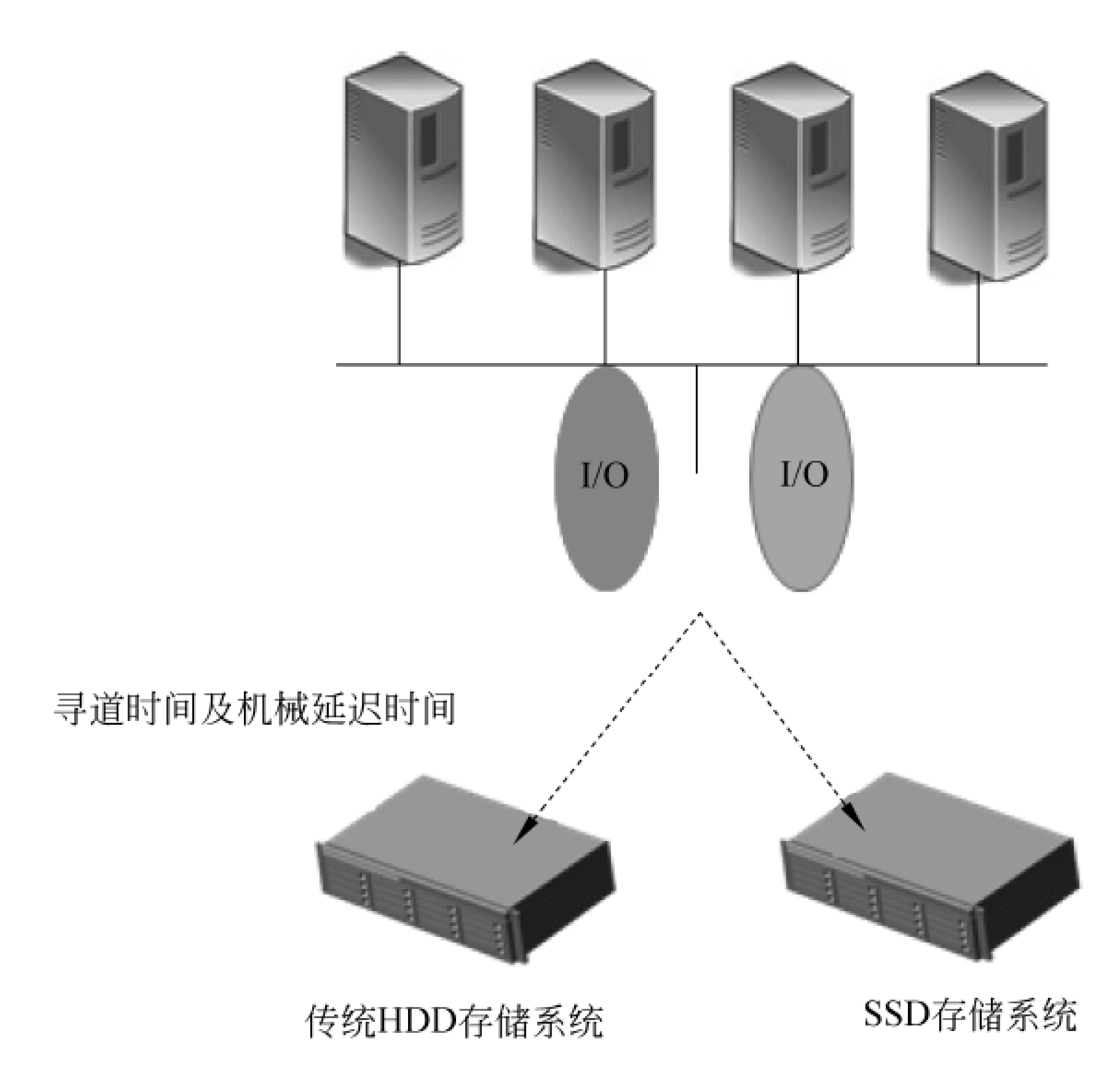
1. 响应时间短

传统硬盘的机械特性导致大部分时间浪费在寻道和机械延迟上,数据传输效率受到严重制约。而SSD硬盘内部没有机械运动部件,省去了寻道时间和机械延迟,可更快捷地响应读写请求。

2. 读写效率高

机械硬盘在进行随机读写操作时,磁头不停地移动,导致读写效率低下,而SSD通过内部控制器计算出数据的存放位置,并进行读写操作,省去了机械操作时间,大大提高了读写效率。4KB随机读写情况下:FC硬盘的性能为400/400 IOPS,SSD硬盘的性能为26000/5600 IOPS,如图2-29所示。

SSD不含高速旋转的机械结构部件,可经得住严苛的环境考验。以华为SSD硬盘为例:HSSD可承受振动加速度16.4G,机械硬盘一般为0.5G以下。HSSD抗冲击1500G,机械硬盘一般为70G左右。



(a) SSD技术性能优势

图 2-29　SSD性能优势

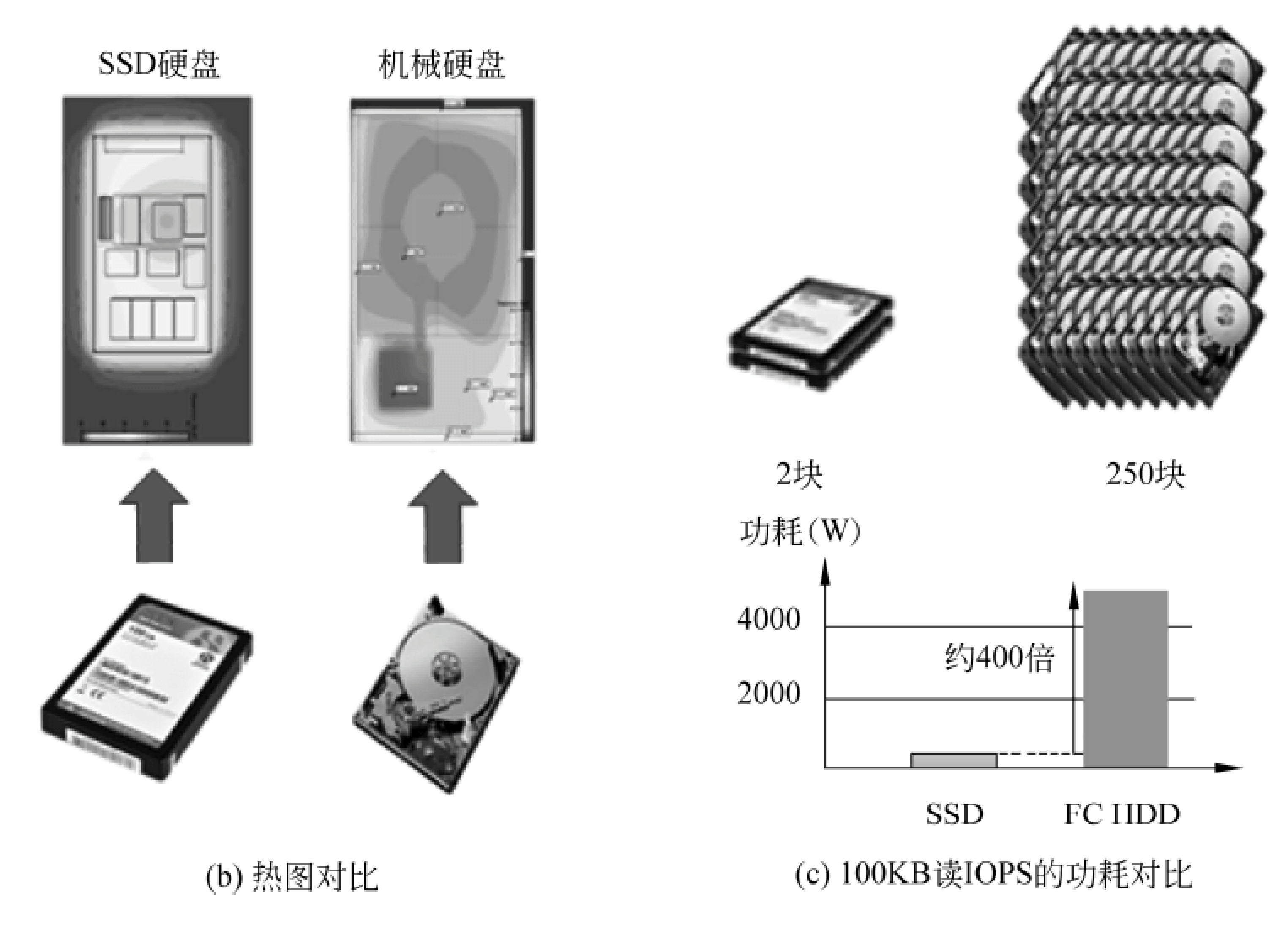


图 2-29　(续)

HSSD 使用专用设备做过如下测试：静压试验、跌落试验、随机振动试验、冲击试验、碰撞试验。

SSD 可用在一些环境较恶劣的场合，如高温高湿、强震等恶劣环境下使用，比如很多工业级应用要求 SSD 固态硬盘做到－20～70℃和－40～85℃的宽温要求。

2.3.5　SSD 在存储中的应用

(1) 二八原则

用户需要频繁改动或读写的数据，一般占存储总量的 20%，称为热数据，对应于 A 级应用。

(2) 分级存储

将热数据存放在 SSD 上，B 级和 C 级应用的数据存放在高速 HDD 和一般 HDD 上，提升性能并减少投资，如图 2-30 所示。

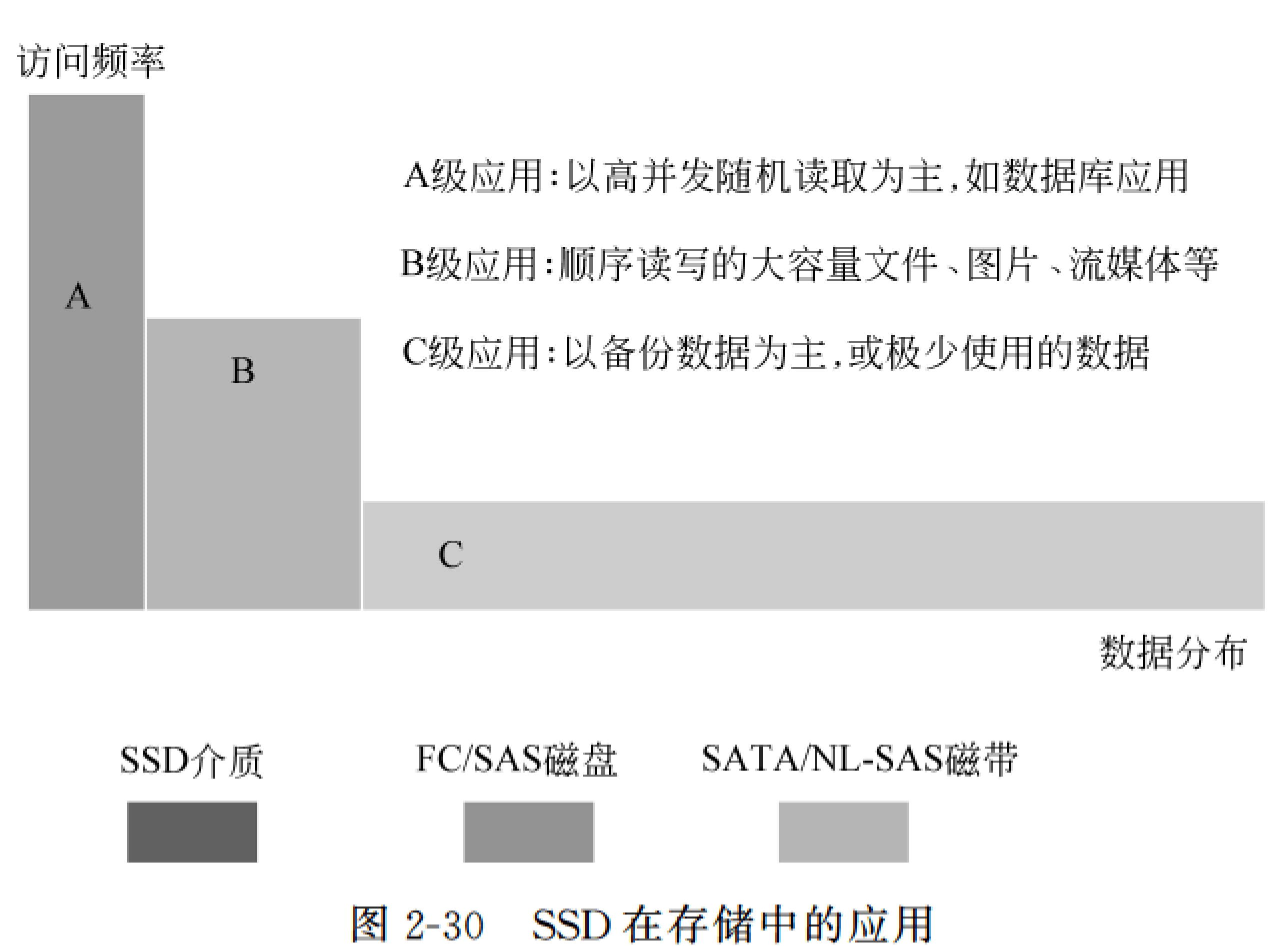


图 2-30　SSD 在存储中的应用

2.4.1　主机硬件系统

主机的核心组件由CPU、存储设备、I/O设备三个部分组成，这三个部分通过总线互联并通信，如图2-31所示。

1. CPU

CPU的运作原理可分为四个阶段：提取(Fetch)、解码(Decode)、执行(Execute)和写回(Write Back)。CPU从存储器或高速缓冲存储器中取出指令，放入指令寄存器，对指令译码，执行指令，如图2-32所示。

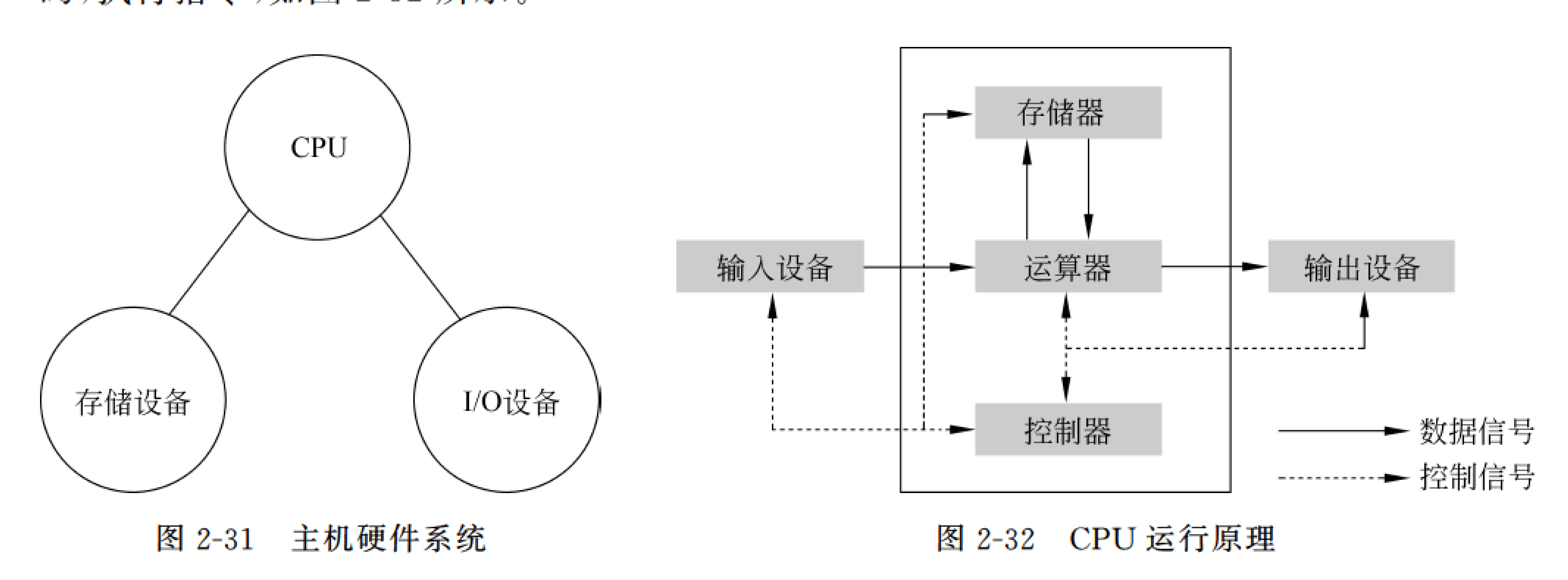


图2-31　主机硬件系统　　　图2-32　CPU运行原理

(1) CISC指令集

CISC(Complex Instruction Set Computer)指令集也称为复杂指令集。英特尔生产的X86系列CPU及其兼容CPU，如AMD、VIA都是属于CISC的范畴。由于Intel X86系列及其兼容CPU都使用X86指令集，所以就形成了今天庞大的X86系列及兼容CPU阵容。

(2) RISC指令集

RISC(Reduced Instruction Set Computing)即“精简指令集”。它是在CISC指令系统基础上发展起来的。RISC指令系统更加适合高档服务器的操作系统UNIX，Linux也属于类似UNIX的操作系统。RISC型CPU与Intel和AMD的CPU在软件和硬件上都不兼容。在中高档服务器中采用RISC指令的CPU主要有以下几类：PowerPC处理器、SPARC处理器、PA-RISC处理器、MIPS处理器、Alpha处理器。

(3) EPIC指令集

EPIC(Explicitly Parallel Instruction Computers，精确并行指令计算机)可以说是Intel的处理器迈向RISC体系的重要步骤。Intel采用EPIC技术的服务器CPU是安腾，也是IA-64系列中的第一款。微软也已开发了64位的操作系统，在软件上加以支持。

2. 内存分类

SDR(synchronous)指同步动态随机存取存储器，这种内存的特性是在一个内存时钟周期中，在一个方波上升沿时进行一次操作(读或写)，一般有两个缺口。

DDR(Double Data Rate)是SDRAM的更新换代产品，允许在时钟脉冲的上升沿和下

降沿传输数据，这样不需要提高时钟的频率就能加倍提高 SDRAM 的速度。DDR 内存不向后兼容 SDRAM。

DDR2 采用和 DDR1 内存一样的指令，工作频率比 DDR1 更高，DDR2 的缺口比较靠中间。

DDR3 与 DDR2 的基础架构并没有本质的不同，DDR3 能提供更高的外部数据传输率，以及更先进的地址/命令与控制总线的拓扑架构，在保证性能的同时将能耗进一步降低。

目前普及的是 DDR3 内存，主要参数有容量、主频、带宽。

3. 总线

总线是计算机内部各个组件之间或者在不同的计算机之间进行数据传输的公共通路，如图 2-33 所示。

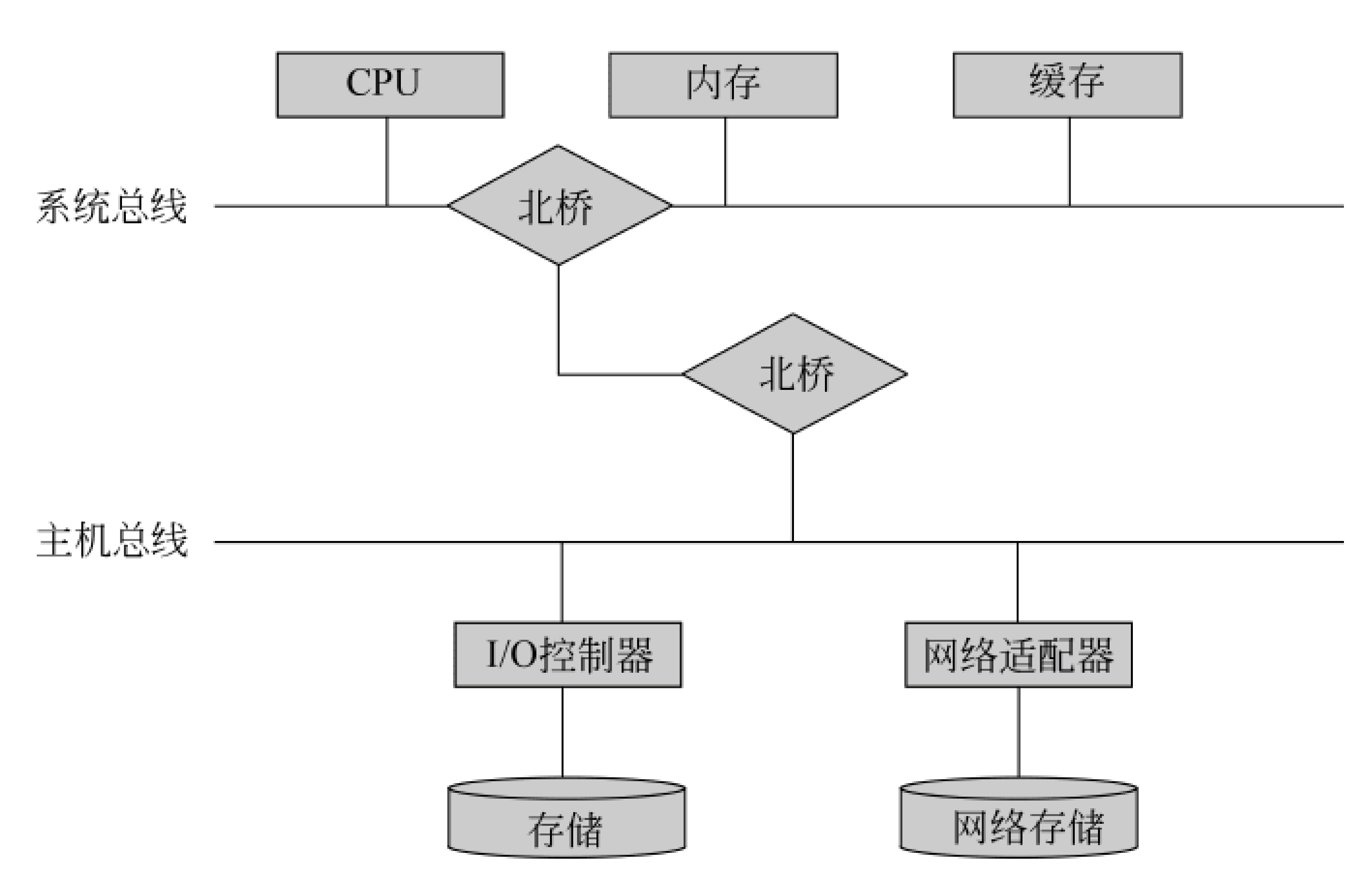


图 2-33　计算机之间进行数据传输

CPU 通过系统总线读取内存中的指令，通过指令的计算结果对内存进行读写操作，CPU 和内存之间通过高速的北桥芯片互联。

主机总线指一些适配器连接到南桥的总线，比如显卡、网卡、RAID 卡、FC HBA 卡等。

I/O 总线指适配器与外部设备相连的总线，比如硬盘、U 盘、存储器等。

4. PCI-E 总线

Intel 在 2001 年春季正式公布了旨在取代 PCI 总线的第三代 I/O 技术，最后却被正式命名为 PCI-Express，Express 意思是高速、特别快的意思。

2002 年 7 月 23 日，PCI-SIG 正式公布了 PCI Express 1.0 规范，并且根据开发蓝图，在 2006 年的时候正式推出 PCI Express 2.0 规范。

经历过三代半（AGP 总线只是一种增强型的 PCI 总线）的发展，PC 的外部总线终于发展到现在的 PCI-E 2.0，提供了比以往总线大得多的带宽。

目前的主流总线是 X8 和 X16。

5. PCI-E 的优势

（1）点对点独享带宽

PCI 是所有设备共享同一条总线资源，PCI-E 采用点对点技术，芯片之间用接口连线，设备之间用数据电缆。

（2）双通道，高速率

类似于全双工模式，速度大大提升，1.0 版本的 PCI-E 在每个信道单方向 2.5Gbps 的传输速率作为起步，物理上可达到 1～32 个可选信道。

（3）热插拔热交换

PCI-E 总线接口插槽中含有“热插拔检测信号”，所以可以像 USB 总线那样进行热插拔和热交换。

（4）多点虚拟通道

PCI-E 总线技术在每一个物理通道中也支持多点虚拟通道，理论上每一个单物理通道中可以有 8 条虚拟通道进行独立的通信控制，而且每个通信的数据包都可以定义不同的 QoS。

（5）兼容 PCI

以前的 PCI 可以在 PCI-E 这一模式下运行，为用户提供了一个平滑的升级平台，但要注意的是不兼容目前的 AGP 接口。

2.4.2 主机软件系统

1. 逻辑卷管理器

在主机软件系统中，操作系统上层的应用软件用于满足用户不同领域、不同业务的应用需求，操作系统实现对计算机硬件与软件资源的管理，并提供用户与系统交互的操作界面。不同的操作系统使用其各自的文件系统，文件系统建立在逻辑卷的基础上。

（1）逻辑卷

逻辑卷改变了数据的存储方式，使数据的存储更具有灵活性，一个逻辑卷可以跨越多个物理磁盘，数据存储在物理上可以是不连续的。逻辑卷建立在卷组之上，卷组中的空间可以建立多个逻辑卷，并且逻辑卷可以随意从卷组的空闲空间中增减，逻辑卷可以属于一个卷组，也可以属于不同的多个卷组。

（2）卷组

卷组建立在物理卷之上，一个卷组中可以包含一个物理卷组或者多个物理卷。

（3）物理卷

物理卷在逻辑卷管理器中属于最底层，任何的逻辑卷和卷组都必须依靠物理卷来建立，物理卷可以是一个完整的硬盘，也可以是硬盘中的一个分区。

2. 文件系统

文件系统是指把文件存储于硬盘时所必须使用的数据结构及硬盘数据的管理方式，如图 2-34 所示。

为了访问硬盘中的数据，就必须在扇区之间建立联系，也就是需要一种逻辑上的数据存储结构。建立这种逻辑结构就是文件系统要做的事情，在硬盘上建立文件系统的过程通常称为“格式化”。

硬盘数据的管理通过文件分区表记录数据的地址，然后通过地址记录实现对数据的读取。

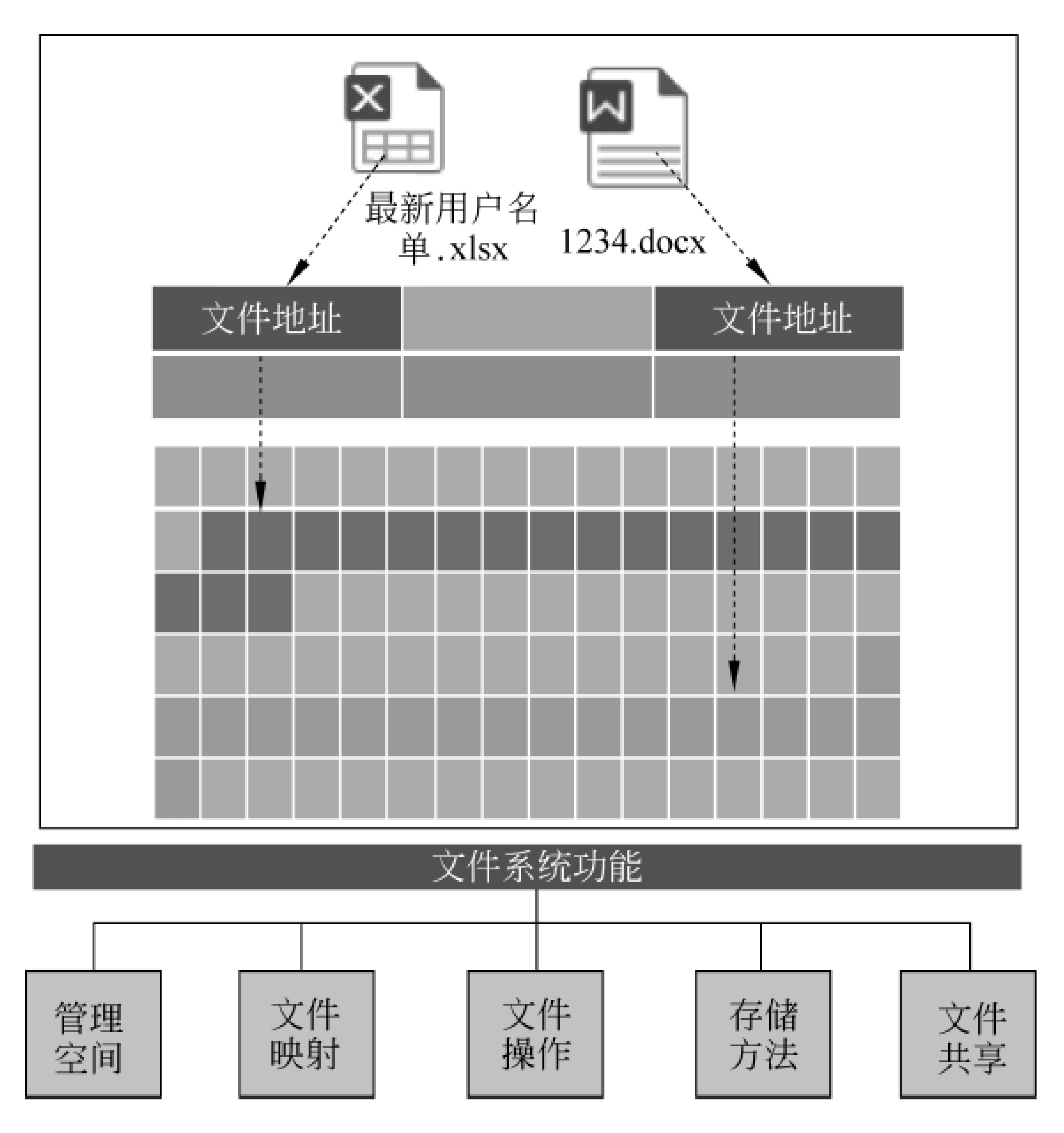


图 2-34　文件系统

3. 操作系统的主要功能

(1) 进程管理

管理进程间的通信,进程异常终止以及进程死锁等。

(2) 内存管理

用于寻找可用内存空间,并配置与释放内存空间等。

(3) 网络通信

提供通信协议的协商。

(4) 安全机制

操作系统提供外界直接或间接使用资源的管道,提供一定的安全机制来控制不同人员的使用权限。

(5) 用户界面

提供图形化的界面,方便人机交互。

(6) 驱动程序

通过驱动程序来屏蔽各厂商设备的差异,操作系统提供统一的接口来管理设备。

4. 主机存储应用

如图 2-35 所示,左边为主机内部的存储 I/O。大部分 I/O 开始于要访问数据的应用,应用通常不管存储的细节,而是直接调用由操作系统提供的系统调用接口,然后由文件系统为数据提供逻辑地址和在磁盘上存储的物理地址的映射,再通过设备驱动层,主要是 SCSI 协议的操作,将数据存储到硬盘上。

图 2-35 的右边为主机通过网络的存储 I/O。首先 I/O 由应用发起,然后经过操作系统,由文件系统提供数据的逻辑地址和存储的物理地址的对应关系,经由设备驱动,到达

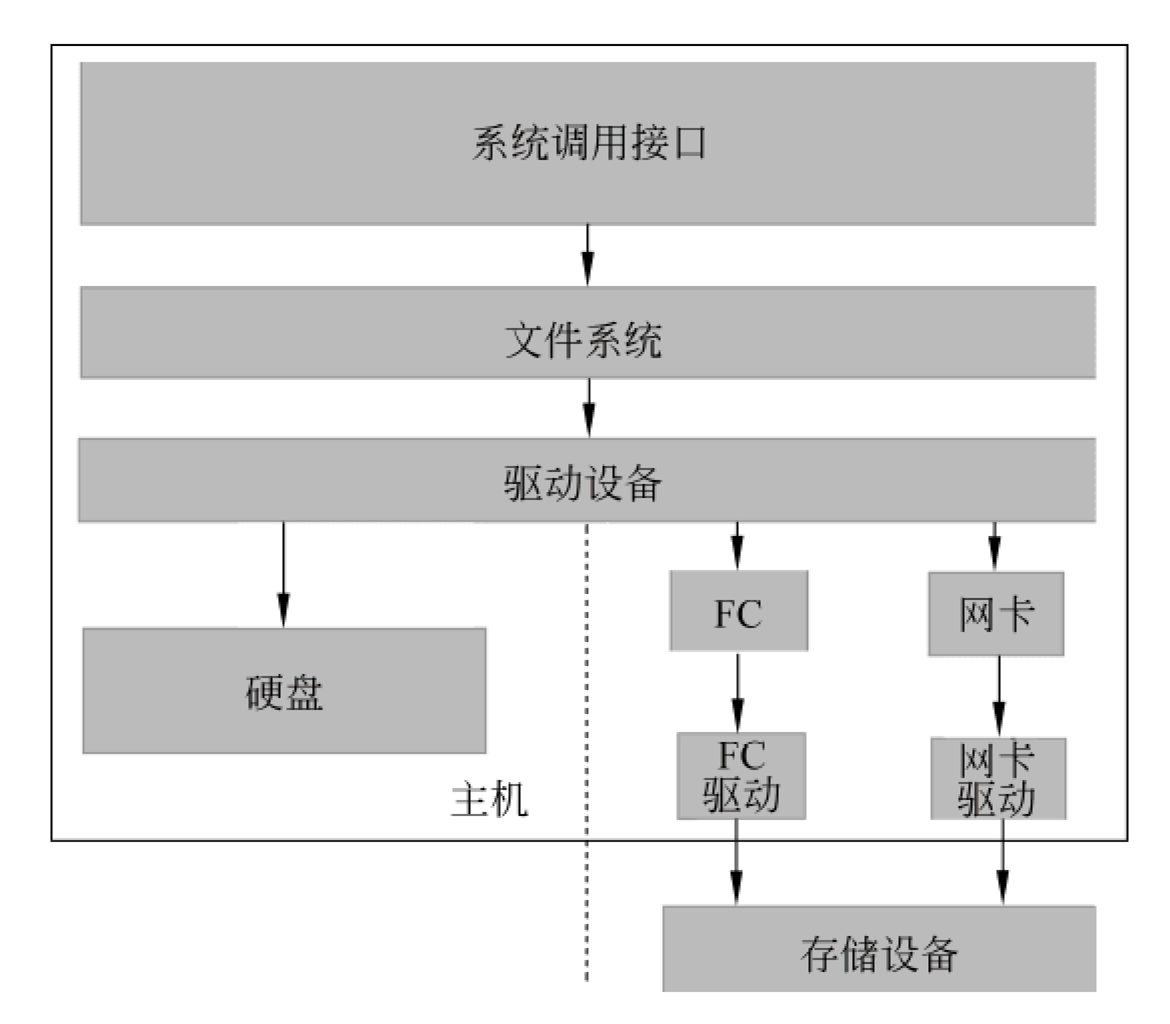


图 2-35　主机存储应用

FC HBA 卡或者网卡，到达存储端的 FC 接口或者网络接口，将数据存储到存储设备上。

2.4.3　主机的管理

一般的主机管理方式为本地管理，而服务器大多数运行在专用的机房或者数据中心，为了实现对存放在非本地工作环境的服务器远程管理，开发了 IPMI 技术，通过 IPMI 技术可远程连接至服务器并对服务器进行管理。

智能平台管理接口(IPMI) 是一种开放标准的硬件管理接口规格，定义了嵌入式管理子系统进行通信的特定方法。IPMI 信息通过基板管理控制器(BMC) 进行交流。使用低级硬件智能管理而不使用操作系统进行管理。

基于 Intel 结构的企业系统中所使用的外围设备采用了一种工业标准，该标准由英特尔、惠普、NEC、美国戴尔电脑和 SuperMicro 等公司制定。用户可以利用 IPMI 监视服务器的物理健康特征，如温度、电压、风扇工作状态、电源状态等。而且更为重要的是 IPMI 是一个开放的免费标准，用户无须为使用该标准而支付额外的费用。

1998 年 Intel、DELL、HP 及 NEC 共同提出 IPMI 规格，可以透过网络远端控制温度、电压。

2001 年 IPMI 从 1.0 版改版至 1.5 版，新增 PCI Management Bus 等功能。

2004 年 Intel 发表了 IPMI 2.0 的规格，能够向下相容 IPMI 1.0 及 1.5 的规格。新增了 Console Redirection，可以通过 Port、Modem 以及 Lan 远端管理伺服器，并加强了安全、VLAN 和刀锋伺器的支援性。

IPMI 是智能型平台管理接口(Intelligent Platform Management Interface)的缩写，是管理基于 Intel 结构的企业系统中所使用的外围设备采用的一种工业标准。

1. IPMI 的工作原理

IPMI 的核心是一个专用芯片/控制器(叫作服务器处理器或基板管理控制器(BMC))，其并不依赖于服务器的处理器、BIOS 或操作系统来工作，是一个单独在系统内运行的无代

理管理子系统。IPMI 良好的自治特性便克服了以往基于操作系统的管理方式所受的限制，例如操作系统不响应或未加载的情况下其仍然可以进行开关机、信息提取等操作，如图 2-36 所示。

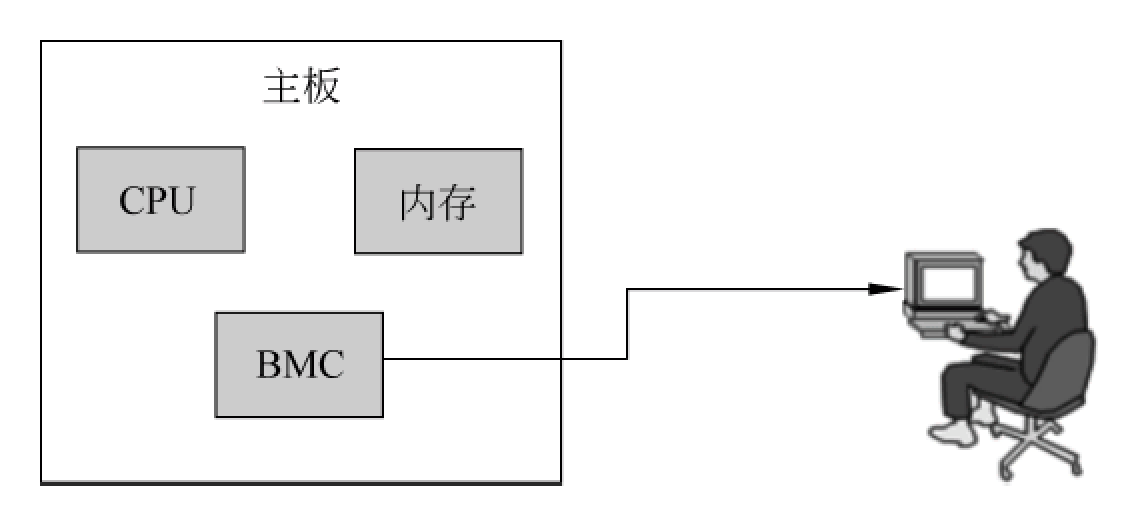


图 2-36　IPMI 的工作原理

在工作时，所有的 IPMI 功能都是向 BMC 发送命令来完成的，命令使用 IPMI 规范中规定的指令，BMC 接收并在系统事件日志中记录事件消息，维护描述系统中传感器情况的传感器数据记录。在需要远程访问系统时，IPMI 新的 LAN 上串行(SOL)特性很有用。SOL 改变 IPMI 会话过程中本地串口的传送方向，从而提供对紧急管理服务、Windows 专用管理控制台或 Linux 串行控制台的远程访问。BMC 通过在 LAN 上改变传送给串行端口的信息的方向来做到这点，提供了一种与厂商无关的远程查看启动、操作系统加载器或紧急管理控制台来诊断和维修故障的标准方式。

2. 查看 BMC IP

在不知道 BMC 管理 IP 的情况下，可以进入 BIOS 查看，不同的服务器主板位置可能会不一样，例如在华为设备下，一般在以下位置，如图 2-37 所示。

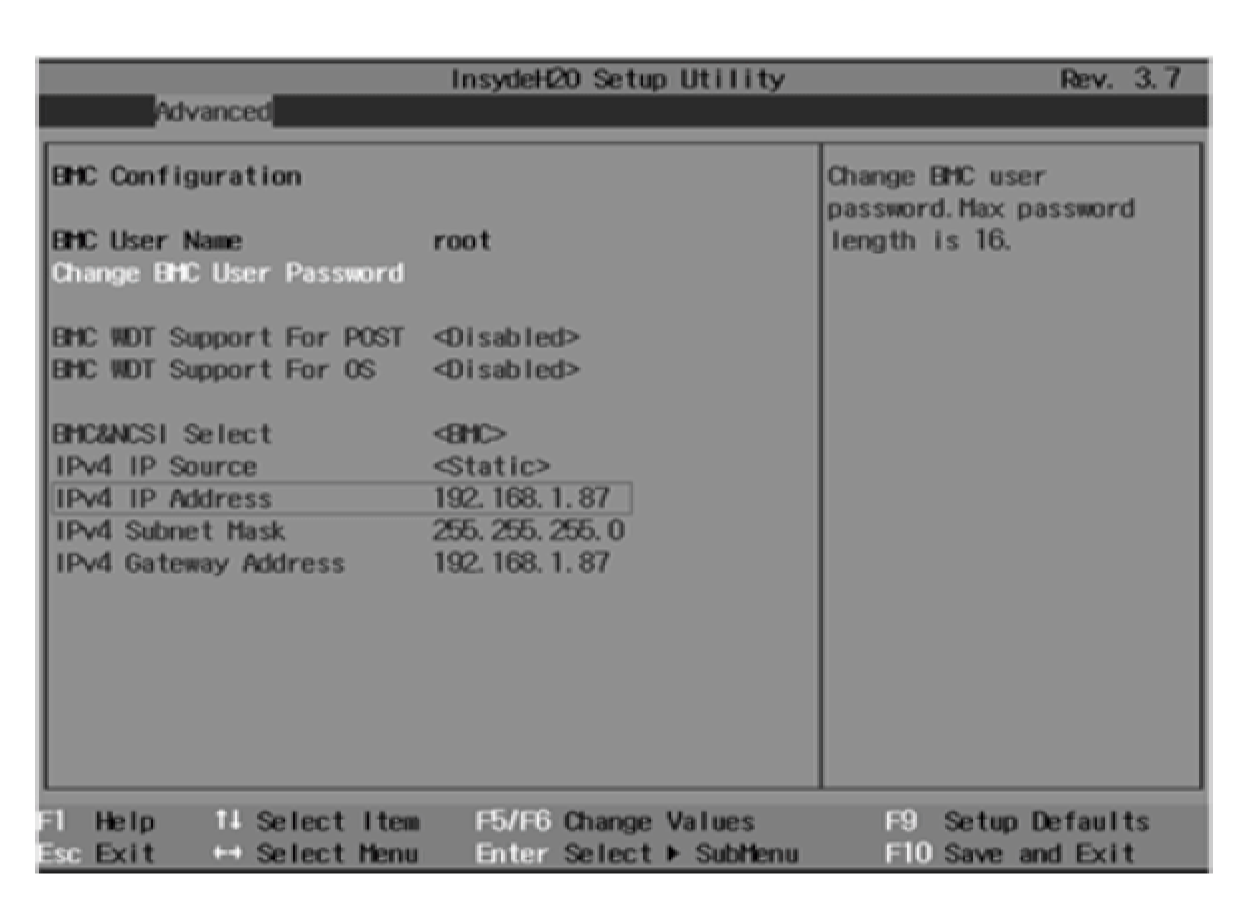


图 2-37　设备的 BMC 信息

(1) 选择 Advanced→IPMI BMC Configuration 命令，按 Enter 键，进入 IPMI BMC Configuration 界面。

(2) 选择 BMC Configuration，按 Enter 键，进入 BMC Configuration 界面，显示 BMC IP 信息。

3. 服务器系统启动选项

系统第一启动设备选项信息包括硬盘、光驱、软驱/可拔插移动设备、PXE(Pre-boot Execution Environment)及未配置。该设置为一次生效,系统在下次启动后该设置将失效,如图 2-38 所示。



图 2-38　服务器系统启动

(1) 硬盘:表示强制从硬盘启动系统。

(2) 光驱:表示强制从 CD/DVD 启动系统。

(3) 软驱/可拔插移动设备:表示强制从软驱或可拔插移动设备上启动系统。

(4) PXE:表示强制从 PXE 上启动系统。

(5) 未配置:表示不进行强制设置,按 BIOS 默认方式启动系统,如图 2-39 所示。

存储系统从来都不是一个对的、封闭的系统。存储系统始终是为应用系统提供存储服务,依赖于应用系统存在的子系统。应用系统通常通过服务器对用户提供服务。服务器是指在网络环境下运行相应的应用软件,为网络用户提供信息资源共享和各种服务的一种高性能计算机。

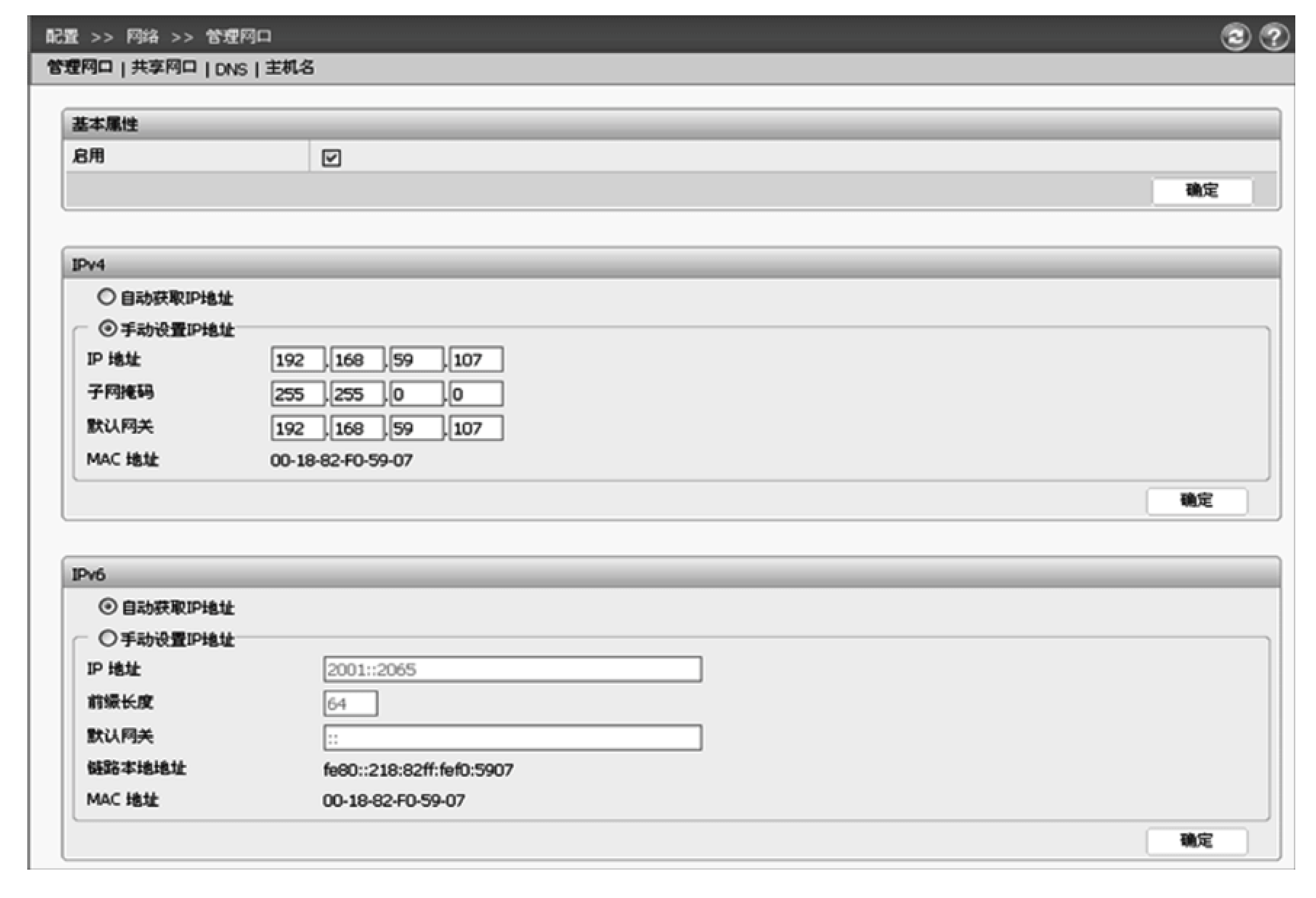


图 2-39　启动服务器系统



图 2-39　（续）

数据库(Database)是按照数据结构来组织、存储和管理数据的仓库,它产生于 1950 年,随着信息技术和市场的发展,特别是 20 世纪 90 年代以后,数据管理不再仅仅是存储和管理数据,而转变成用户所需要的各种数据管理的方式。数据库有很多种类型,从最简单的存储各种数据的表格到能够进行海量数据存储的大型数据库系统都在各个方面得到了广泛的应用。

在信息化社会,充分有效地管理和利用各类信息资源,是进行科学研究和决策管理的前提条件。数据库技术是管理信息系统、办公自动化系统、决策支持系统等各类信息系统的核心部分,是进行科学研究和决策管理的重要技术手段。

2.5.1　数据库的基本概念

数据库,简单来说是本身可视为电子化的文件柜——存储电子文件的处所,用户可以对文件中的数据进行新增、查询、更新、删除等操作。

数据库指的是以一定方式储存在一起、能被多个用户共享、具有尽可能小的冗余度、与应用程序彼此独立的数据集合。

在经济管理的日常工作中,常常需要把某些相关的数据放进这样的“仓库”,并根据管理的需要进行相应的处理。

例如,企业或事业单位的人事部门常常要把本单位职工的基本情况(职工号、姓名、年龄、性别、籍贯、工资、简历等)存放在表中,这张表就可以看成是一个数据库。有了这个“数据仓库”就可以根据需要随时查询某职工的基本情况,也可以查询工资在某个范围内的职工人数等。这些工作如果都能在计算机上自动进行,那我们的人事管理就可以达到极高的水平。此外,在财务管理、仓库管理、生产管理中也需要建立众多的这种“数据库”,使其可以利用计算机实现财务、仓库、生产的自动化管理。

数据库是依照某种数据模型组织起来并存放二级存储器中的数据集合。这种数据集合

具有如下特点：尽可能不重复，以最优方式为某个特定组织的多种应用服务，其数据结构独立于使用它的应用程序，对数据的增、删、改、查由统一软件进行管理和控制。从发展的历史看，数据库是数据管理的高级阶段，它是由文件管理系统发展起来的。

2.5.2 数据库处理系统

数据库是一个单位或是一个应用领域的通用数据处理系统，它存储的是属于企业和事业部门、团体和个人的有关数据的集合。数据库中的数据是从全局观点出发建立的，按一定的数据模型进行组织、描述和存储。其结构基于数据间的自然联系，从而可提供一切必要的存取路径，且数据不再针对某一应用，而是面向全组织，具有整体的结构化特征。

数据库中的数据是为众多用户所共享其信息而建立的，已经摆脱了具体程序的限制和制约。不同的用户可以按各自的用法使用数据库中的数据；多个用户可以同时共享数据库中的数据资源，即不同的用户可以同时存取数据库中的同一个数据。数据共享性不仅满足了各用户对信息内容的要求，同时也满足了各用户之间信息通信的要求。

2.5.3 数据库的基本结构

数据库的基本结构分三个层次，反映了观察数据库的三种不同角度。

(1) 物理数据层

它是数据库的最内层，是物理存储设备上实际存储的数据的集合。这些数据是原始数据，是用户加工的对象，由内部模式描述的指令操作处理的位串、字符和字组成。

(2) 概念数据层

它是数据库的中间一层，是数据库的整体逻辑表示。指出了每个数据的逻辑定义及数据间的逻辑联系，是存储记录的集合。它所涉及的是数据库所有对象的逻辑关系，而不是它们的物理情况，是数据库管理员概念下的数据库。

(3) 用户数据层

它是用户所看到和使用的数据库，表示了一个或一些特定用户使用的数据集合，即逻辑记录的集合。

数据库不同层次之间的联系是通过映射进行转换的。

2.5.4 数据库的主要特点

(1) 实现数据共享

数据共享包含所有用户可同时存取数据库中的数据，也包括用户可以用各种方式通过接口使用数据库，并提供数据共享。

(2) 减少数据的冗余度

同文件系统相比，由于数据库实现了数据共享，从而避免了用户各自建立应用文件。减少了大量重复数据，减少了数据冗余，维护了数据的一致性。

(3) 数据的独立性

数据的独立性包括逻辑独立性(数据库的逻辑结构和应用程序相互独立)和物理独立性(数据物理结构的变化不影响数据的逻辑结构)。

(4) 数据实现集中控制

文件管理方式中,数据处于一种分散的状态,不同的用户或同一用户在不同处理中其文件之间毫无关系。利用数据库可对数据进行集中控制和管理,并通过数据模型表示各种数据的组织以及数据间的联系。

(5) 数据保持一致性和可维护性,以确保数据的安全性和可靠性

主要包括:①安全性控制:以防止数据丢失、错误更新和越权使用;②完整性控制:保证数据的正确性、有效性和相容性;③并发控制:使在同一时间周期内,允许对数据实现多路存取,又能防止用户之间的不正常交互作用。

(6) 故障恢复

由数据库管理系统提供一套方法,可及时发现故障和修复故障,从而防止数据被破坏。数据库系统能尽快恢复数据库系统运行时出现的故障,可能是物理上或是逻辑上的错误。比如对系统的误操作造成的数据错误等。

2.5.5　数据库的结构模型

数据库通常分为层次式数据库、网络式数据库和关系式数据库三种。而不同的数据库是按不同的数据结构来联系和组织的。

1. 数据结构模型

(1) 数据结构

所谓数据结构是指数据的组织形式或数据之间的联系。如果用 D 表示数据,用 R 表示数据对象之间存在的关系集合,则将 $DS=(D,R)$称为数据结构。

例如,设有一个电话号码簿,它记录了 n 个人的名字和相应的电话号码。为了方便地查找某人的电话号码,将人名和号码按字典顺序排列,并在名字的后面跟随着对应的电话号码。这样,若要查找某人的电话号码(假定他的名字的第一个字母是 Y),那么只需查找以 Y 开头的那些名字就可以了。该例中,数据的集合 D 就是人名和电话号码,它们之间的联系 R 就是按字典顺序排列的,其相应的数据结构就是 $DS=(D,R)$,即一个数组。

(2) 数据结构类型

数据结构又分为数据的逻辑结构和数据的物理结构。

数据的逻辑结构是从逻辑的角度(即数据间的联系和组织方式)来观察数据,分析数据,与数据的存储位置无关;数据的物理结构是指数据在计算机中存放的结构,即数据的逻辑结构在计算机中的实现形式,所以物理结构也被称为存储结构。

这里只研究数据的逻辑结构,并将反映和实现数据联系的方法称为数据模型。

比较流行的数据模型有三种,即按图论理论建立的层次结构模型和网状结构模型以及按关系理论建立的关系结构模型。

2. 层次、网状和关系数据库系统

(1) 层次结构模型

层次结构模型实质上是一种有根节点的定向有序树(在数学中“树”被定义为一个无回的连通图)。比如,高等学校的组织结构图像一棵树,校部就是树根(称为根节点),各系、专业、教师、学生等为枝点(称为节点),树根与枝点之间的联系称为边,树根与边之比为 1∶N,即树根只有一个,树枝有 N 个。

按照层次模型建立的数据库系统称为层次模型数据库系统。IMS(Information Management System)是其典型代表。

(2) 网状结构模型

按照网状数据结构建立的数据库系统称为网状数据库系统,其典型代表是DBTG(Database Task Group)。用数学方法可将网状数据结构转化为层次数据结构。

(3) 关系结构模型

关系式数据结构把一些复杂的数据结构归结为简单的二元关系(即二维表格形式)。例如某单位的职工关系就是一个二元关系。

由关系数据结构组成的数据库系统被称为关系数据库系统。

在关系数据库中,对数据的操作几乎全部建立在一个或多个关系表格上,通过对这些关系表格的分类、合并、连接或选取等运算来实现数据的管理。

因此,可以概括地说,一个关系称为一个数据库,若干个数据库可以构成一个数据库系统。数据库系统可以派生出各种不同类型的辅助文件和建立它的应用系统。

2.5.6 数据库的备份与恢复

Oracle数据库有三种标准的备份方法,它们分别是导出/导入(EXP/IMP)、热备份和冷备份。导出备件是一种逻辑备份,冷备份和热备份是物理备份。

1. 导出/导入(Export/Import)

利用Export可将数据从数据库中提取出来,利用Import则可将提取出来的数据送回到Oracle数据库中去。

(1) 简单导出数据(Export)和导入数据(Import)

Oracle支持以下三种类型的输出:

① 表方式(T方式),将指定表的数据导出;

② 用户方式(U方式),将指定用户的所有对象及数据导出;

③ 全库方式(Full方式),将数据库中的所有对象导出。

数据导入(Import)的过程是数据导出(Export)的逆过程,分别将数据文件导入数据库和将数据库数据导出为数据文件。

(2) 增量导出/导入

增量导出是一种常用的数据备份方法,它只能对整个数据库来实施,并且必须作为SYSTEM来导出。在进行此种导出时,系统不要求回答任何问题。导出文件名默认为export.dmp。如果不希望自己的输出文件定名为export.dmp,必须在命令行中指出要用的文件名。

(3) 增量导出包括的三种类型如下。

① "完全"增量导出(Complete)即备份三个数据库,比如:

```
exp system/manager inctype=complete file=040731.dmp
```

② "增量型"增量导出备份上一次备份后改变的数据,比如:

```
exp system/manager inctype=incremental file=040731.dmp
```

③ "累积型"增量导出累计型导出方式是导出自上次"完全"导出之后数据库中变化了

的信息。比如：

```
exp system/manager inctype=cumulative file=040731.dmp
```

数据库管理员可以排定一个备份日程表，用数据导出的三个不同方式合理高效地完成。比如数据库的导出任务可以做如下安排。

星期一：完全备份(A)；

星期二：增量导出(B)；

星期三：增量导出(C)；

星期四：增量导出(D)；

星期五：累计导出(E)；

星期六：增量导出(F)；

星期日：增量导出(G)。

如果在星期日，数据库遭到意外破坏，数据库管理员可按以下步骤来恢复数据库。

第一步：用 CREATE DATABASE 命令重新生成数据库结构；

第二步：创建一个足够大的附加回滚；

第三步：完全增量导入 A，即：

```
imp system/manager inctype=RESTORE FULL=y FILE=A
```

第四步：累计增量导入 E：

```
imp system/manager inctype=RESTORE FULL=Y FILE=E
```

第五步：最近增量导入 F：

```
imp system/manager inctype=RESTORE FULL=Y FILE=F
```

2. 冷备份

冷备份发生在数据库已经正常关闭的情况下，当正常关闭时会提供给我们一个完整的数据库。冷备份是将关键性文件复制到另外位置的一种说法。对于备份 Oracle 信息而言，冷备份是最快和最安全的方法。冷备份的优点如下。

- 是非常快速的备份方法(只需复制文件)。
- 容易归档(简单复制即可)。
- 容易恢复到某个时间点上(只需将文件再复制回去)。
- 能与归档方法相结合，做数据库“最佳状态”的恢复。
- 低度维护，高度安全。

但冷备份也有如下不足。

- 单独使用时，只能提供到“某一时间点上”的恢复。
- 在实施备份的全过程中，数据库必须要做备份而不能做其他工作。也就是说，在冷备份过程中，数据库必须处于关闭状态。
- 若磁盘空间有限，只能复制到磁带等其他外部存储设备上，速度会很慢。
- 不能按表或按用户恢复。

如果可能(主要看效率)，应将信息备份到磁盘上，然后启动数据库(使用户可以工作)并

将备份的信息复制到磁带上(复制的同时,数据库也可以工作)。冷备份中必须复制的文件包括:

①所有数据文件;②所有控制文件;③所有联机 REDO LOG 文件;④Init. ora 文件(可选)。

值得注意的是冷备份必须在数据库关闭的情况下进行,当数据库处于打开状态时,执行数据库文件系统的备份是无效的。

下面是做冷备份的完整例子。

(1) 关闭数据库

```
sqlplus/nolog
sql> connect/as sysdba
sql> shutdown normal;
```

(2) 用复制命令备份全部的时间文件、重做日志文件、控制文件、初始化参数文件

```
sql> cp
```

(3) 重启 Oracle 数据库

```
sql> startup
```

3. 热备份

热备份是在数据库运行的情况下,采用归档模式(archivelog mode)方式备份数据库的方法。所以,如果用户有昨天夜里做过一个冷备份而且又有今天的热备份文件,在发生问题时,就可以利用这些资料恢复更多的信息。热备份要求数据库在 Archivelog 方式下操作,并需要大量的档案空间。一旦数据库运行在 archivelog 状态下,就可以做备份了。热备份的命令文件由以下三部分组成。

(1) 数据文件是一个表空间和一个表空间的备份

① 设置表空间为备份状态;

② 备份表空间的数据文件;

③ 恢复表空间为正常状态。

(2) 备份并归档 log 文件

① 临时停止归档进程;

② log 下 archive rede log 目标目录中的文件;

③ 重新启动 archive 进程;

④ 备份归档的 redo log 文件。

(3) 用 alter database backup controlfile 命令来备份控制文件

热备份的优点是:

- 可在表空间或数据库文件级备份,备份的时间短;
- 备份时数据库仍可使用;
- 可达到秒级恢复(恢复到某一时间点上);
- 可对几乎所有数据库实体做恢复;
- 恢复是快速的,大多数情况下在数据库仍工作时恢复。

热备份的不足是：

- 不能出错，否则后果严重；
- 若热备份不成功，所得结果不可用于时间点的恢复；
- 因难于维护，所以要特别仔细小心，不允许“以失败告终”。

任务拓展

1. 存储数据中心的基础设施与中心机房构建。
2. 目前存储介质在数据存储中的应用。
3. 主机软硬件系统的基本架构与管理。
4. 数据库在存储应用系统方面的具体应用案例。

第3章　数据存储技术



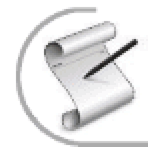
任务目标

- 熟悉数据存储阵列系统的组成；
- 掌握数据存储阵列系统技术；
- 掌握数据存储阵列基础配置；
- 掌握 SAN 技术原理以及应用；
- 掌握 IP SAN 技术原理以及应用；
- 掌握 SNAS 技术原理以及应用。



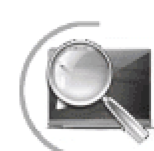
项目背景

企业存储技术发展日新月异，早期大型服务器的 DAS 技术（Direct Attached Storage，直接附加存储，又称直连存储），后来为了提高存储空间的利用率及管理安装上的效率，因而有了 SAN（Storage Area Network，存储局域网络）技术的诞生，传统存储阵列不能像软件定义那样适应最新的业务应用，成本高，操作复杂，数据访问慢，受 RAID 组态变化的影响较大。SAN 可以说是 DAS 网络化发展趋势下的产物。SAN 采用的是光纤通道（FC，Fiber Channel）技术，所以在 iSCSI 出现以前，SAN 多半单指 FC 而言。一直到 iSCSI 出现，分为 FC-SAN 及 iSCSI-SAN 存储。紧接着，为了能在多用户网络环境中做好档案集中化分享管理的工作，采用全然不同于以往的文件协议（File Protocol）数据存取方式的 NAS（Network Attached Storage，网络附加存储）方案也应运而生。互联网的日益发展，IP 存储成为最佳存储方案，随着大数据、云计算的发展，云存储的出现省去了本地维护环节，把数据都存在了别处。闪存和混合存储不受磁盘阵列延迟的影响而快速访问数据。融合存储融合了服务器和存储，对象存储用分布式存储代替了中心化存储，所有新思路的提出都是为了解决传统存储阵列所面临的问题。



项目描述

聚比特科技有限公司建立网络基础平台建作为信息化支持中心、应用推广中心和公司技术管理数据中心，根据企业数据应用特点，综合考虑可用性、性能和成本来解决传统存储阵列所面临的问题，制定当前最佳存储技术方案。



项目分析

聚比特科技有限公司数据中心的业务数据分析。数据呈现爆炸性增长，不仅对存储的

性能及容量提出苛刻考验,还要具备快速的数据检索与分析能力以及时获取关键价值信息,所用存储技术方案是解决问题的核心。



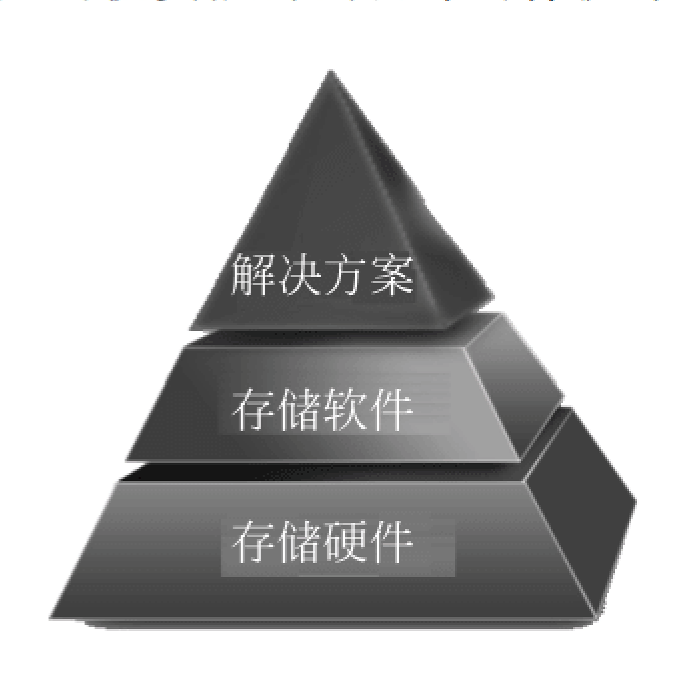
项目实现

面对公司业务数据分析,数据呈现爆炸性增长,活跃数据归档需要更加简便,同时又得考虑成本和存储方案的有效性。美国市场研究公司 IDC 也认为“在可预见的未来,存储是大数据和分析领域最大的基础设施开支之一”。所以针对目前公司业务数据现状,选择适合当前公司业务发展的存储方案,再逐步升级完善。

3.1　存储阵列系统的组成

3.1.1　存储阵列系统的基本概念

当今的存储技术不是一个单独而孤立的技术,实际上,完整的存储系统应该是由一系列组件构成。目前,人们把存储系统分为了硬件架构部分、软件组件部分以及实际应用时的存储解决方案部分。硬件部分又包括外置的存储系统,主要是指人们实际的存储设备,比如磁盘阵列、磁带库等。因为软件组件的存在,使存储设备的可用性得到了大大的提高,数据的镜像、复制,自动的数据备份等数据操作都可以通过对存储软件的控制来完成。一个设计良好的存储解决方案,是使数据存储工作更加简单易行的最佳保障,设计优秀的存储解决方案,不仅可以使存储系统实际部署的时候更简单容易,更可以降低客户的总体拥有成本(TCO),使客户的投资能得到良好的保护,如图 3-1 所示。



- 容灾解决方案
- 备份解决方案
- ……

- 存储管理软件 (ISM)
- 快照、镜像软件
- 备份软件

- 外置存储系统
  磁盘阵列
  NAS
  磁带库
  虚拟磁带库
- 存储连接设备
  FC HBA 卡
  FC 交换机
  以太网交换机
  连接线缆

图 3-1　存储阵列系统

3.1.2　存储阵列在存储系统架构中的位置

在存储系统架构中,磁盘阵列作为数据存储设备,为用户业务系统提供数据存储服务。存储阵列设备成为用户数据业务稳定、可靠、高效运行的重要因素,如图 3-2 所示。

3.1.3　存储阵列系统硬件的组成

存储阵列系统主要由控制框和硬盘框两部分组成,为客户提供一个高可靠、高性能、大容量的智能化存储系统,如图 3-3 所示。

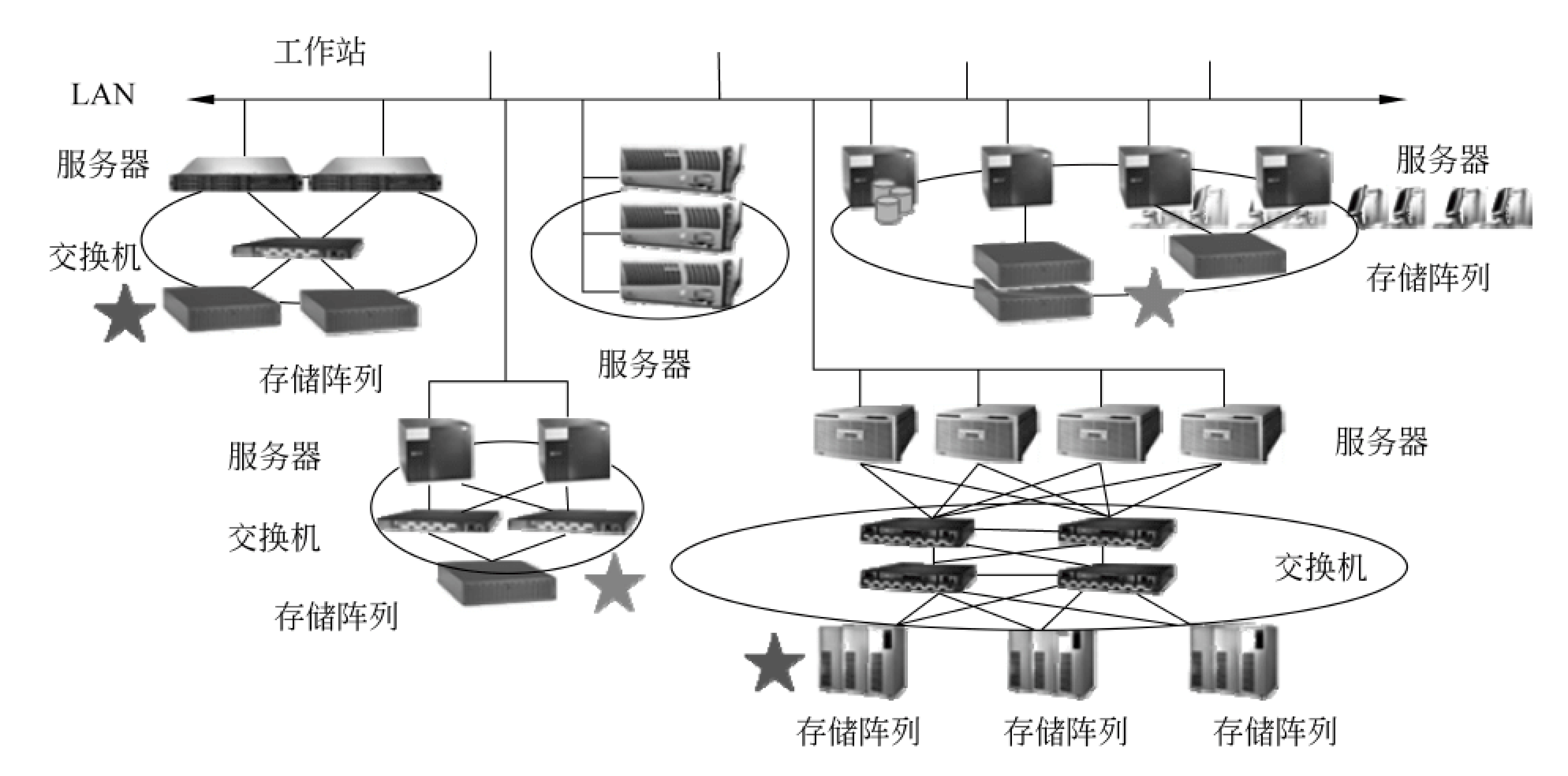


图 3-2 存储系统构架



图 3-3 存储阵列系统硬件

(1) 控制框：控制框用于处理各种存储业务，并管理级联在控制框下面的硬盘框。

(2) 硬盘框：硬盘框主要用于容纳各种硬盘，为应用服务器提供充足的存储空间。

3.2 传统的存储系统

3.2.1 传统的内置存储

在传统的计算机存储系统中，存储工作通常是由计算机内置的硬盘来完成的，而采用这样的设计方式，硬盘本身的缺陷很容易成为整个系统的性能瓶颈，并且由于机箱内有限的空间限制了硬盘数量的扩展，同时也对机箱内的散热、供电等提出了严峻的挑战，再加上不同的计算机各自为战，使用各自内置的硬盘，导致从总体看来存储空间的利用率较低，并且分散保存的数据也不利于数据的共享和备份工作。在传统的 C/S 架构中，无论使用的是何种协议，存储设备都直接与服务器相连接。在这样的结构下，对存储设备上所保存的所有数据的任何读写操作，都必须由服务器来进行，这样的处理方式给服务器带来了沉重的负担。外部存储系统的出现，彻底将服务器从烦琐的 I/O 操作中解放出来，使服务器更加专门化，使之仅仅承担应用数据的操作任务，得以更充分地释放自身潜能。把存储设备从服务器中分离出来，使它们变成直接与网络连接的网络存储设备。所以存储区域网中，存储设备不再属于哪个应用服务器，从而可以对存储设备实施集中管理，使用户可以方便地共享存储资源。为网络上的应用系统提供丰富的存储资源及快速、简便的访问方式，如图 3-4 所示。

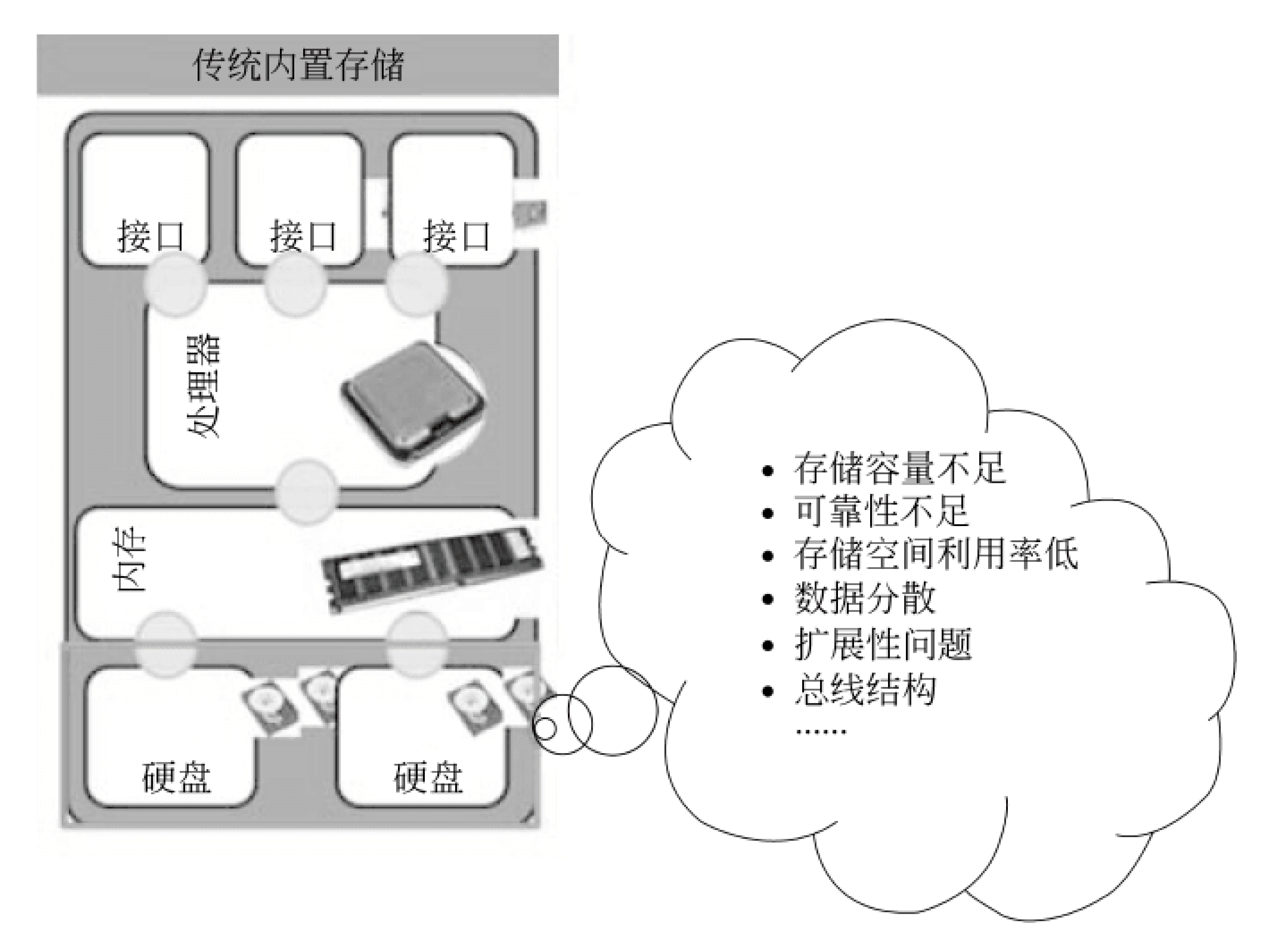


图 3-4　传统内置存储

存储网络建立了独立的基于网络的存储架构，增强了现有的计算拓扑架构。存储网络允许存储设备直接连接到现有网络上，也可以通过专门的存储网络进行连接，这一技术给传统的存储配置方案带来了两个重要的变化。

（1）存储网络在存储设备、服务器以及客户机之间建立了更多的直接访问路径，从而使用户事务能够绕过大量的服务器 I/O 操作而直接与数据发生联系，从而避免了对服务器进行不必要的访问。

（2）存储网络使得商务应用系统能够以更高的效率访问数据。换言之，存储网络使得应用系统能够更方便地共享数据，并赋予服务器更为强大的数据连接能力。

3.2.2　外置存储网络的形态

（1）直接连接存储（Direct Attached Storage，DAS）。由于早期的网络十分简单，所以直接连接存储得到发展。随着计算能力、内存、存储密度和网络带宽的进一步增长，越来越多的数据被存储在个人计算机和工作站中。分布式的计算和存储的增长对存储技术提出了更高的要求。由于使用 DAS，存储设备与主机的操作系统紧密相连，其典型的管理结构是基于 SCSI 的并行总线式结构。存储共享是受限的，原因是存储是直接依附在服务器上的。从另一方面看，系统也因此背上了沉重的负担，因为 CPU 必须同时完成磁盘存取和应用运行的双重任务，所以不利于 CPU 的指令周期的优化。

（2）网络连接存储（Network Attached Storage，NAS）。局域网在技术上得以广泛实施，在多个文件服务器之间实现了互联，为实现文件共享而建立一个统一的框架。随着计算机应用越来越广泛，大量的不兼容性导致数据的获取日趋复杂。因此采用广泛使用的局域网加工作站族的方法就对文件共享、互操作性和节约成本有很大的意义。NAS 包括一个特殊的文件服务器和存储设备。NAS 服务器上采用优化的文件系统，并且安装了预配置的存储设备。由于 NAS 是连接在局域网上的，所以客户端可以通过 NAS 系统与存储设备交互数据。另外，NAS 直接运行文件系统协议，诸如 NFS、CIFS 等。客户端系统可以通过磁盘

映射和数据源建立虚拟连接。

(3) 存储区域网络(Storage Area Networks,SAN)。一个存储网络是一个用在服务器和存储资源之间的、专用的、高性能的网络体系。它为了实现大量原始数据的传输而进行了专门的优化。因此,可以把 SAN 看成是对 SCSI 协议在长距离应用上的扩展。SAN 使用的典型协议组是 SCSI 和 Fiber Channel。Fiber Channel 特别适合这项应用,原因在于,一方面它可以传输大块数据,另一方面它能够实现远距离传输。SAN 的市场主要集中在高端的、企业级的存储应用上。这些应用对于性能、冗余度和数据的可获得性都有很高的要求。

3.2.3 DAS 存储的形态

1. 外部硬盘阵列

JBOD 即 Just a Bunch Of Disks,其在逻辑上把几个物理磁盘串联在一起,其目的纯粹是为了增加磁盘的容量,并不提供数据安全保障。能够解决内置存储有限硬盘槽位及容量扩展不足的问题。但仍然是基于单硬盘存放数据,可靠性差,如图 3-5 所示。

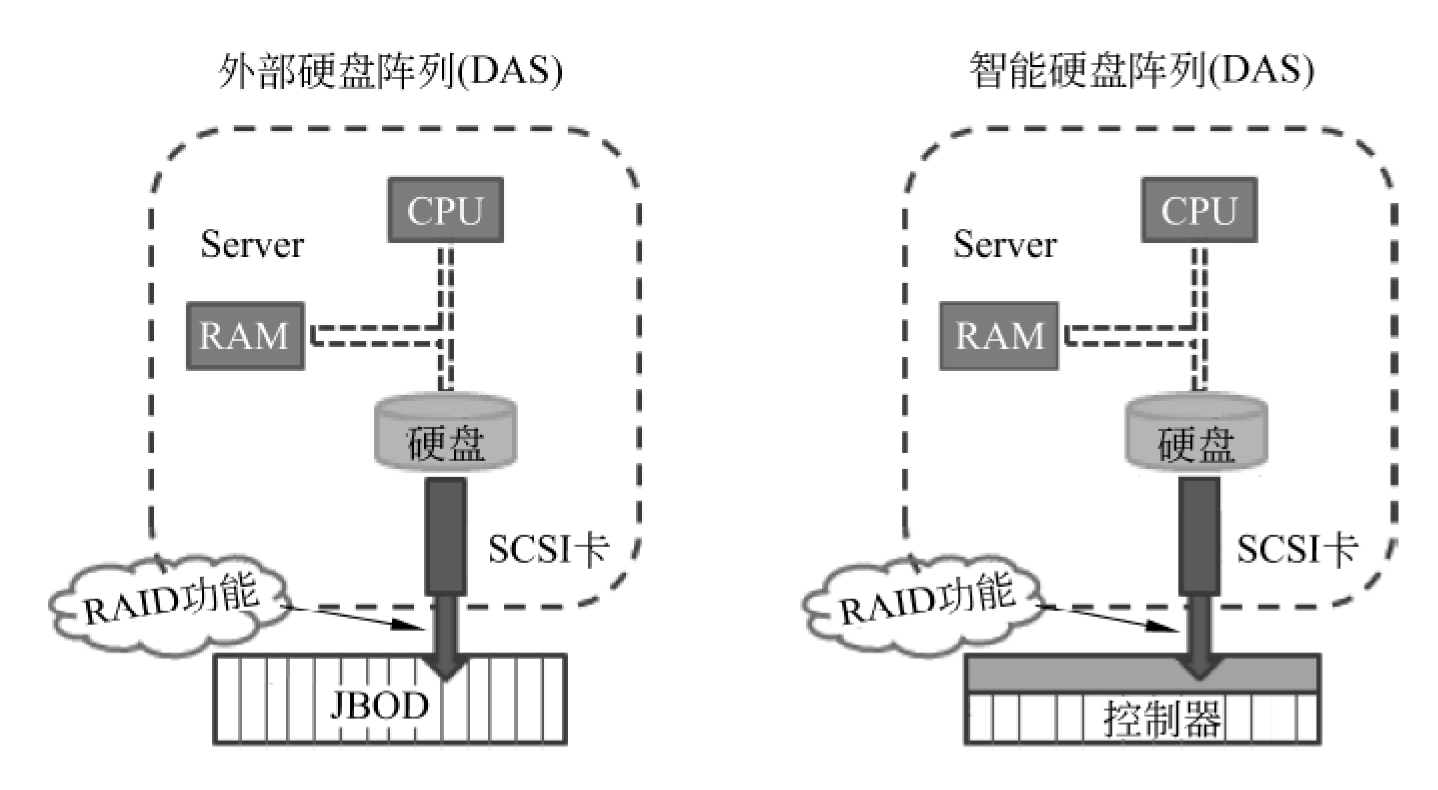


图 3-5 硬盘阵列

2. 智能硬盘阵列

控制器中包含 RAID 功能、大容量 Cache,同时使得磁盘阵列具有多种实用的功能,配置专用管理软件进行配置管理。

3.2.4 DAS 存储的局限性

DAS 存储方式实现了机内存储到存储子系统的跨越,但其也存在很多局限性。

(1) 扩展性差:服务器与存储设备采用直接连接的方式,当需要新增应用服务器时,只能为新增的服务器单独配置存储设备,造成重复投资。

(2) 浪费资源:存储空间无法充分利用,存在浪费。不同的应用服务器面对的存储数据量是不一致的,同时业务发展的状况也决定着存储数据量的变化。因此,出现了一部分应用对应的存储空间不够用,另一部分却有大量的存储空间闲置。

(3) 管理分散:DAS 方式数据依然是分散的,不同的应用各有一套存储设备,管理分散,无法集中。

(4) 异构化严重:DAS 方式使得企业在不同阶段采购了不同型号、不同厂商的存储设备,设备之间异构化现象严重,导致维护成本居高不下。

(5) 数据备份问题：DAS 方式与主机直接连接，在对重要的数据进行备份时，将会极大地占用网络的带宽。

DAS 特别适合于对存储容量要求不高、服务器的数量很少的中小型局域网，其主要的优点在于存储容量扩展的实施非常简单，投入的成本少而见效快。通常使用 SCSI 协议实现主机服务器与存储设备的互联。

3.3.1 SAN 存储基础

1. 存储区域网络概念

存储区域网络(Storage Area Networks，SAN)是一个用在服务器和存储资源之间的、专用的、高性能的网络体系。SAN 是独立于 LAN 的服务器后端存储的专用网络。SAN 采用可扩展的网络拓扑结构连接服务器和存储设备，每个存储设备不隶属于任何一台服务器，所有的存储设备都可以在全部的网络服务器之间作为对等资源共享。

SAN 主要利用 Fibre Channel protocol(光纤通道协议)，通过 FC 交换机建立起与服务器和存储设备之间的直接连接，因此我们通常也称这种利用 FC 连接建立起来的 SAN 为 FC-SAN。FC 特别适合这项应用，原因在于一方面它可以传输大块数据，另一方面它能够实现较远距离传输。SAN 主要应用在对性能、冗余度和数据的可获得性都有很高要求的高端企业级存储上，如图 3-6 所示。

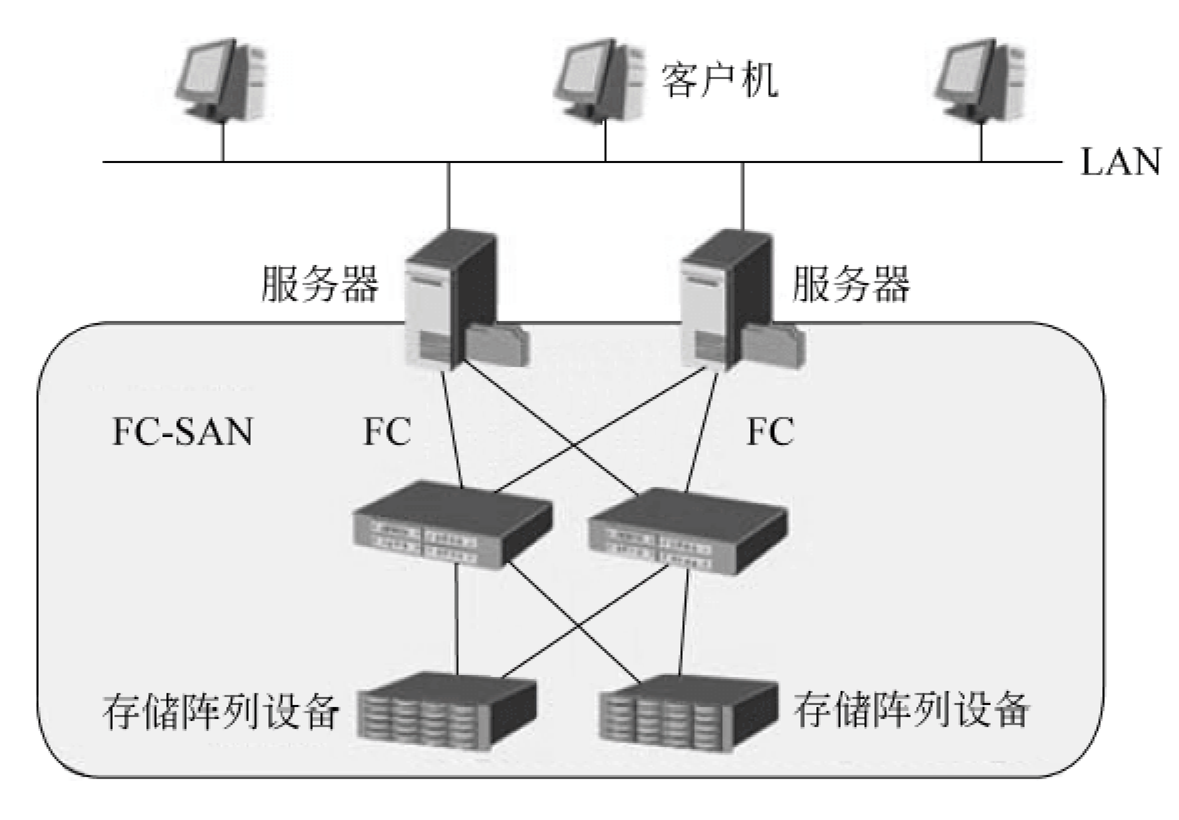


图 3-6 存储区域网络

随着存储技术的发展，目前基于 TCP/IP 协议的 IP-SAN 也得到很广泛的应用。IP-SAN 具备很好的扩展性、灵活的互通性，并能够突破传输距离的限制，具有明显的成本优势和管理维护容易等特点。

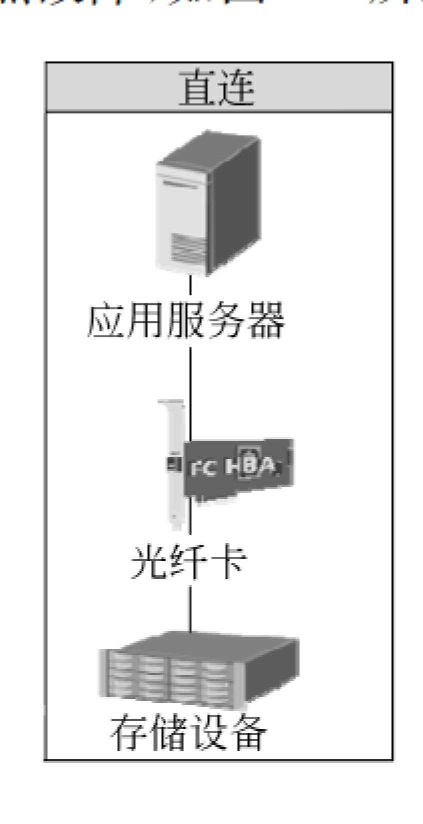
2. SAN 典型组网

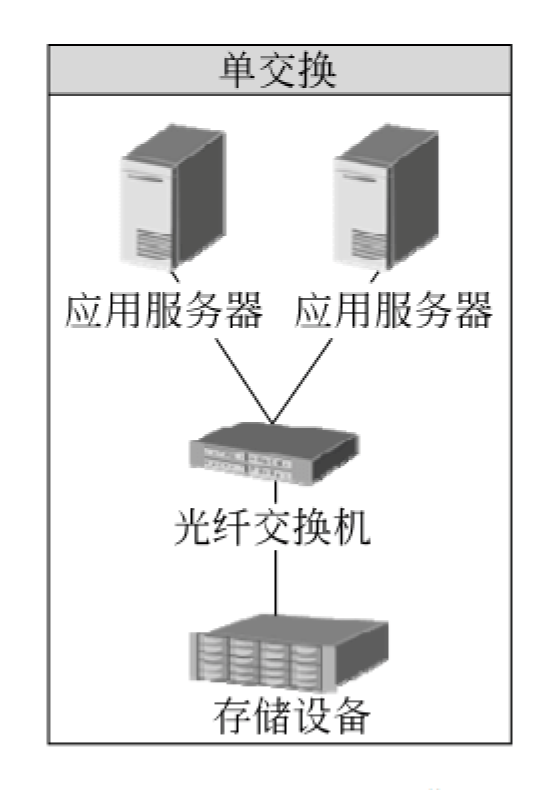
(1) 直连：主机与存储之间通过 FC HBA 卡连接，这种组网方式简单、经济，但较多的主机分享存储资源比较困难。

(2) 单交换：主机与存储之间由一台 FC 交换机连接，这种组网结构使多台主机能共同

分享同一台存储设备,扩展性强,但交换机处存在单点故障。

(3) 双交换:同一台主机到存储阵列端可由多条路径连接,扩展性强,避免了在交换机处形成单点故障,如图 3-7 所示。





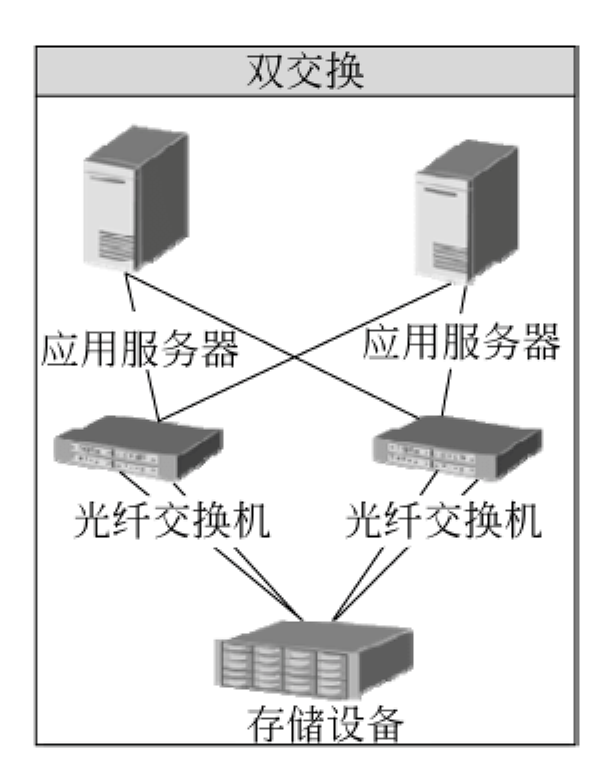


图 3-7 SAN 典型组网

3. SAN 的组件

光纤存储区域网络由四个主要的部分组成,这些组件覆盖了 I/O 操作、存储系统以及所支持的工作负荷的各个主要方面。同时,在 SAN 技术中,还需要一些其他组件进行集成,以构建完整的解决方案。在考虑 SAN 的基础设施建设时,我们需要谨慎地考虑构成 SAN 基础设施的各个不同的组件,因为在 SAN 中,这些组件或者相互独立运作,或者相互依赖以协同工作,如图 3-8 所示。

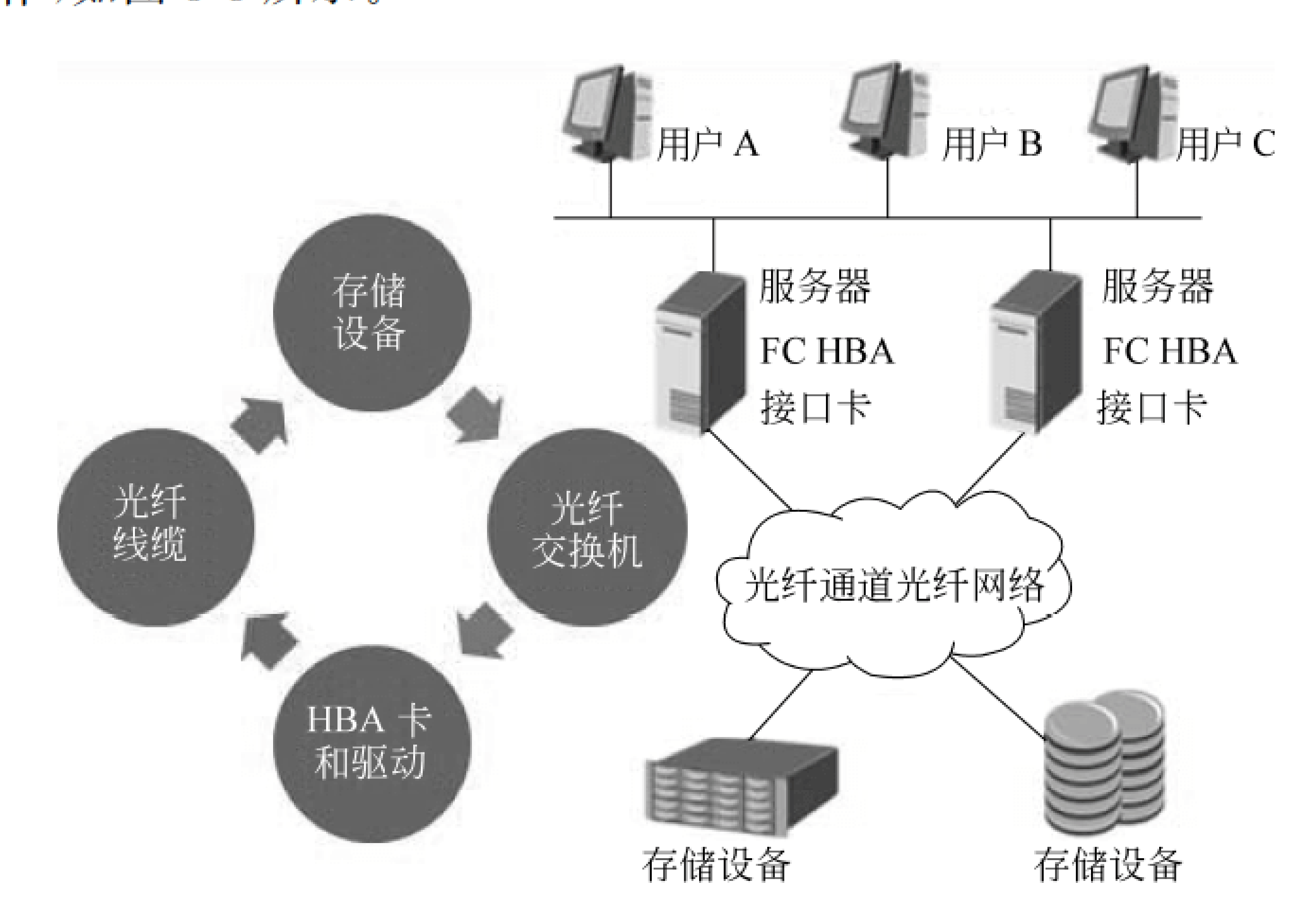


图 3-8 SAN 的组件

4. SAN 存储网络的特点

与传统 DAS 存储相比,SAN 存储网络具备非常明显的优势,如图 3-9 所示。

(1) 基于 FC 光纤介质,拥有千兆位的存储带宽,更适合大容量数据块业务高速处理的要求,目前主流带宽为 8Gbps。

(2) 由于 SAN 存储网络中存储设备(如磁盘阵列,磁带库等)集中部署,可以实现对设



图 3-9　SAN 存储网络的特点

备的集中管理，同时也可以通过远程方式登录设备实现远程管理，管理方式更加灵活。

（3）存储资源集中统一部署，分别映射给各应用服务器，实现存储资源的共享，同时可以根据各应用服务器对存储资源的需求为其动态地分配资源，实现存储资源的动态共享。

（4）在 SAN 存储网络中，数据的传输、复制、迁移、备份等在 SAN 网内高速进行，不需占用业务网络 WAN/LAN 的带宽资源。

（5）在 SAN 存储网络环境下，扩展存储资源变得非常容易，只需要增加新的存储设备到 SAN 存储网络中即可，实现平滑的扩容。新增的存储资源可以直接映射给应用服务器使用。

（6）由于 SAN 存储网络使用的 FC 协议实现了对 SCSI 协议的封装，因此可以实现对以前的各种 SCSI 存储设备的兼容，在异构环境下，更能体现其优势。

5. SAN 存储的应用

SAN 存储网络主要应用在以下方面，如图 3-10 所示。

（1）对响应时间、可用性和可扩展性要求高的关键任务数据库应用。

（2）对性能、数据完整性和可靠性要求高的集中存储备份，以保证关键数据的安全，可极大地提高企业数据备份和恢复操作可靠性和可扩展性。

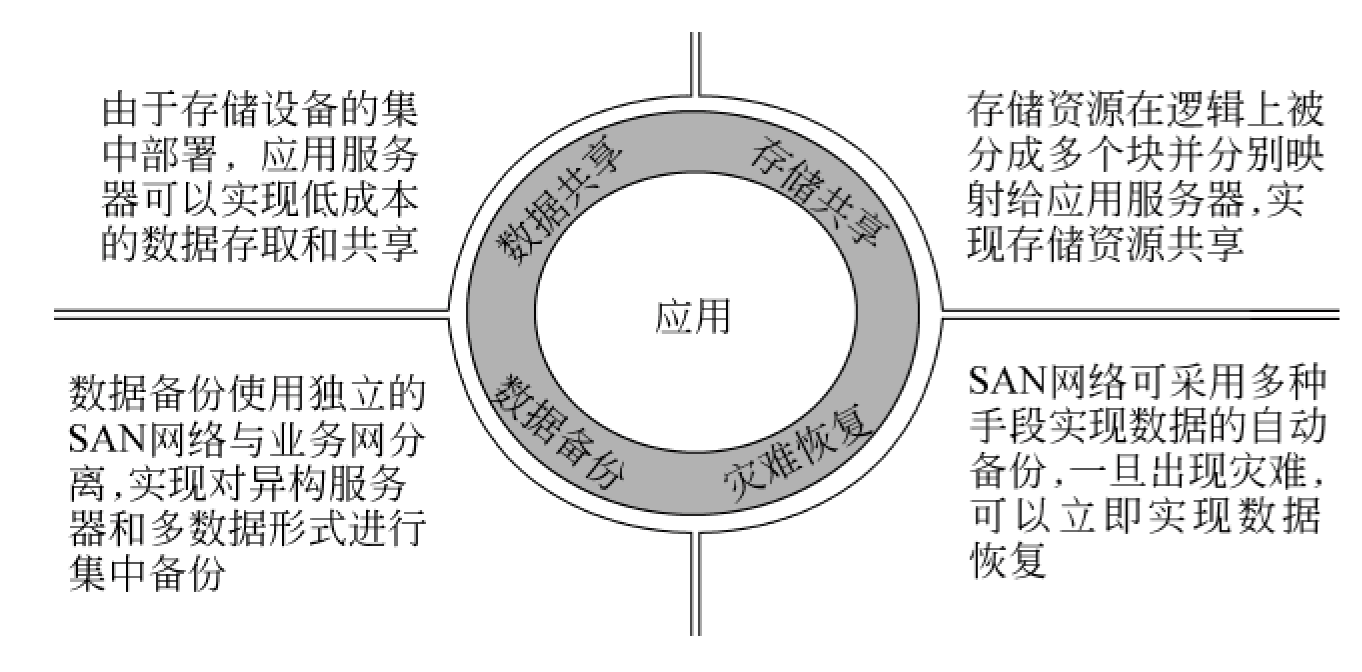


图 3-10　SAN 存储的应用

(3) 海量存储的应用环境。例如，图书馆、银行、证券、中大型企业或组织的数据中心。

(4) 支持服务器及其连接设备之间提供光纤通道高性能和扩展的距离。

3.3.2 FC连接与协议

1. FC拓扑结构

FC主要有三种拓扑结构，用以描述各个节点的连接方式。光纤通道术语中的“节点”是指通过网络进行通信的任何实体，而不一定是一个硬件节点。这个节点通常是一个设备，比如说一个磁盘存储器，服务器上的一个主机总线适配器或者是一个光纤网交换机，如图3-11所示。

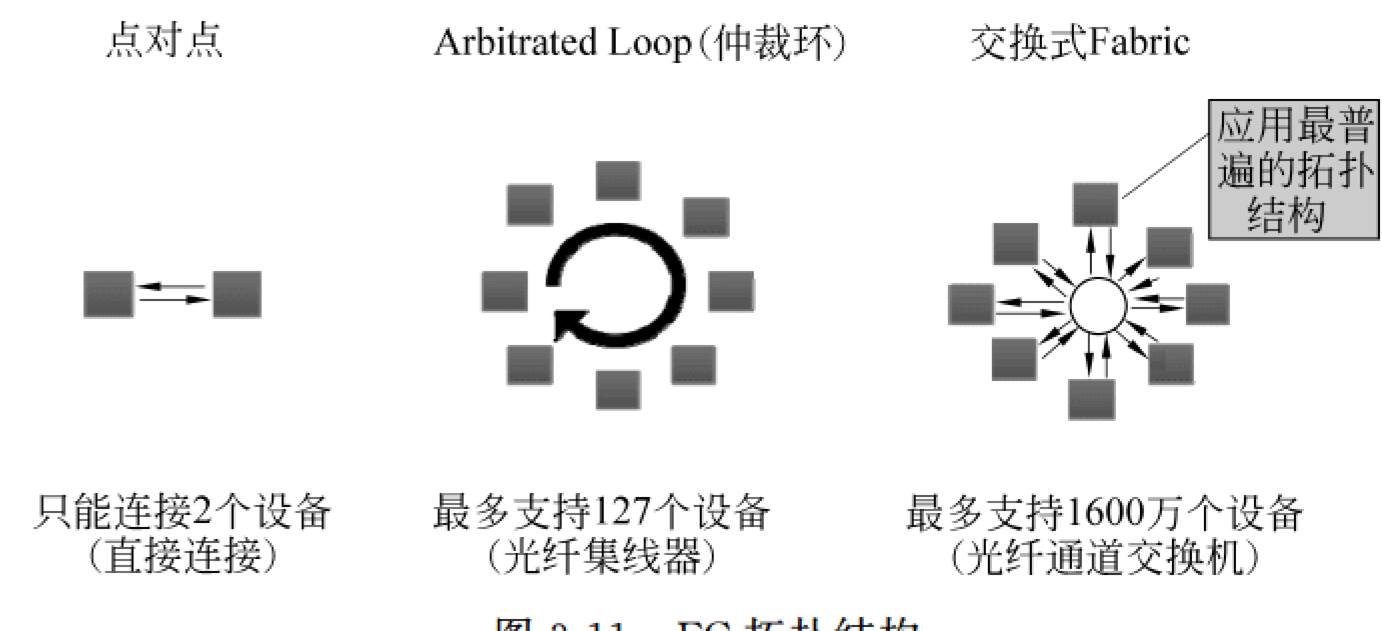


图3-11　FC拓扑结构

(1) 点对点式：两个设备背对背直接连接。这是最简单的一种拓扑，连接能力受限。

(2) 仲裁环式：这种设计方式中，所有设备连接在一个类似于令牌环的环路上。这个环路中添加或者移除一个设备会导致环路上所有活动中断。一个设备的故障导致整个环路不能进行工作。光纤通道集线器能够用于将众多设备连接到一起形成一个逻辑上的环路，并且能够旁路故障节点，使得环上节点的故障不会影响整个环路的通信。一个环路也可以通过使用线缆直接将节点一个接一个地连接成一个环而实现。最小的环路只包含两个节点，这种结构看起来和点到点式连接近似，它们的区别在很大程度上取决于各自的协议。

(3) 光纤交换式：所有的设备或者设备环都被连接到光纤网交换机上，与现有的以太网的实现形式在概念上是类似的。这种拓扑结构相对于点到点和仲裁环的优势在于：

① 交换机对结构形式进行管理，提供了最好的互联形式；

② 多对节点可以同时通信；

③ 各个节点的故障是孤立的，不会危及其他节点的工作。

2. FC协议栈

FC开发于1988年，最早是用来提高硬盘协议的传输带宽，侧重于数据的快速、高效、可靠传输。到20世纪90年代末，FC-SAN开始得到大规模的广泛应用。光纤通道的主要部分实际上是FC-2。其中从FC-0到FC-2被称为FC-PH，也就是“物理层”。光纤通道主要通过FC-2来进行传输，因此，光纤通道也常被称为“二层协议”或者“类以太网协议”。

光纤通道的数据单元称为帧。即使光纤通道本身就有几个层，大部分光纤通道是指第2层协议。一个光纤通道帧最大是2 148字节，而且光纤通道帧的头部比起广域网的IP和TCP来说有些奇怪。光线通道只使用一个帧格式来在多个层上完成各种任务。帧的功能

决定其格式。相比在 IP 世界中的概念，光纤通道帧格式是奇特而且奇妙的。

光纤通道帧起始于帧开始(SOF)标志，接下来是帧头部，数据或光纤通道内容紧随其后，然后是帧结束(EOF)。这样封装的目的是让光纤通道可以在需要时被其他类似于 TCP 的协议所承载，如图 3-12 所示。

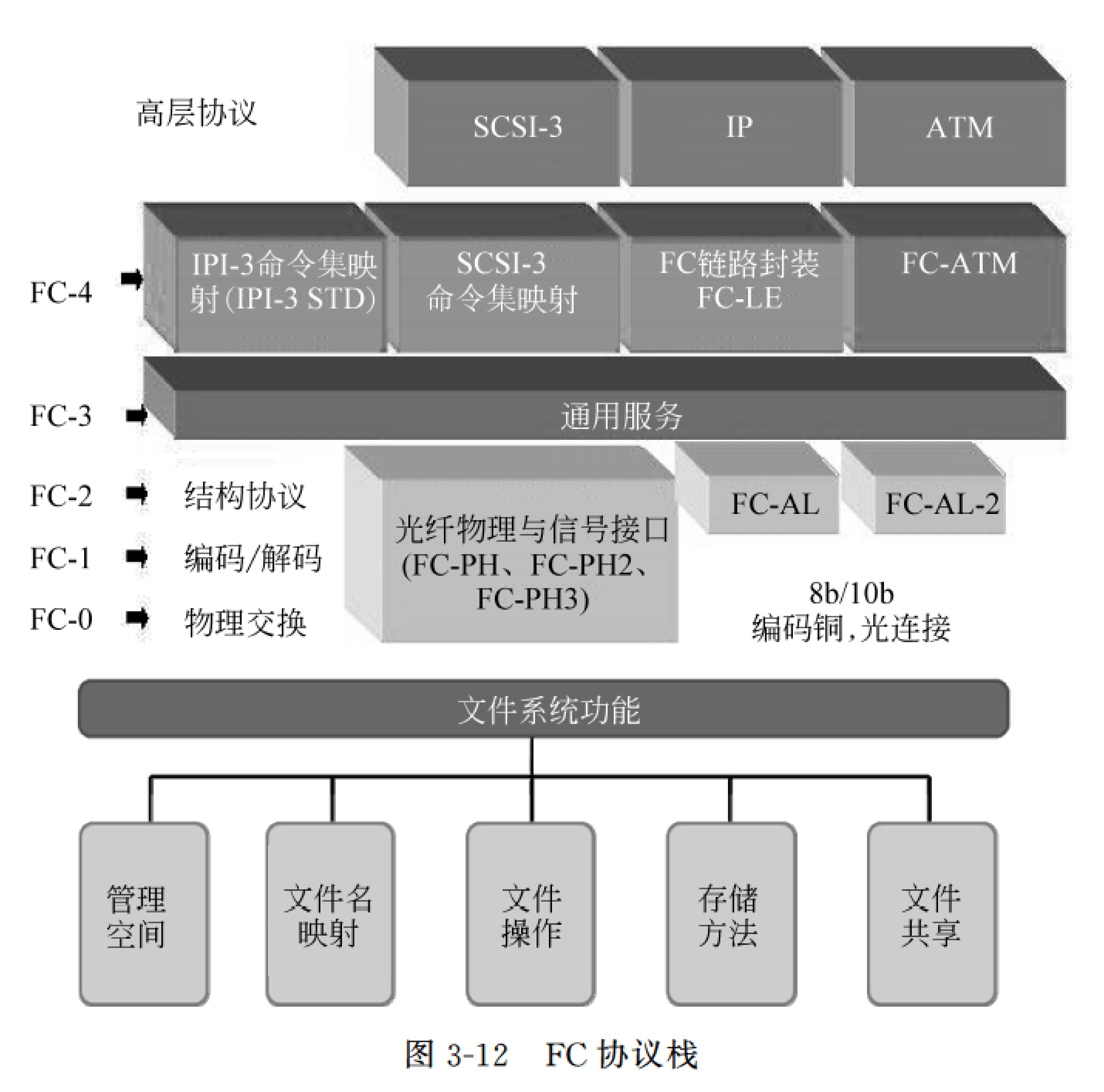


图 3-12　FC 协议栈

FC-0：物理层，定制了不同介质，传输距离，信号机制标准，也定义了光纤和铜线接口以及电缆指标。

FC-1：定义编码和解码的标准。

FC-2：定义了帧、流控制和服务质量等。

FC-3：定义了常用服务，如数据加密和压缩。

FC-4：协议映射层，定义了光纤通道和上层应用之间的接口，上层应用比如串行 SCSI 协议。HBA 的驱动提供了 FC-4 的接口函数。FC-4 支持多协议，如 FCP-SCSI、FC-IP、FC-VI。

3. FC 与 SCSI 协议

FC 与 SCSI 协议的关系如图 3-13 所示。

(1) FC 通道并不是 SCSI 的替代，FC 可以通过构建帧来传输 SCSI 的指令、数据和状态信息单元。

(2) SCSI 是位于光纤通道协议栈 FC4 的上层协议，SCSI 是 FC 协议的子集。

4. FC 存储设备

8Gb FC 接口模块提供传输速率为 8Gb/s 的主机端口。当连接的设备传输速率低于主机端口速率时，主机端口将自动适应传输速率，以保证数据传输通道的连通性和数据传输速率的一致性。

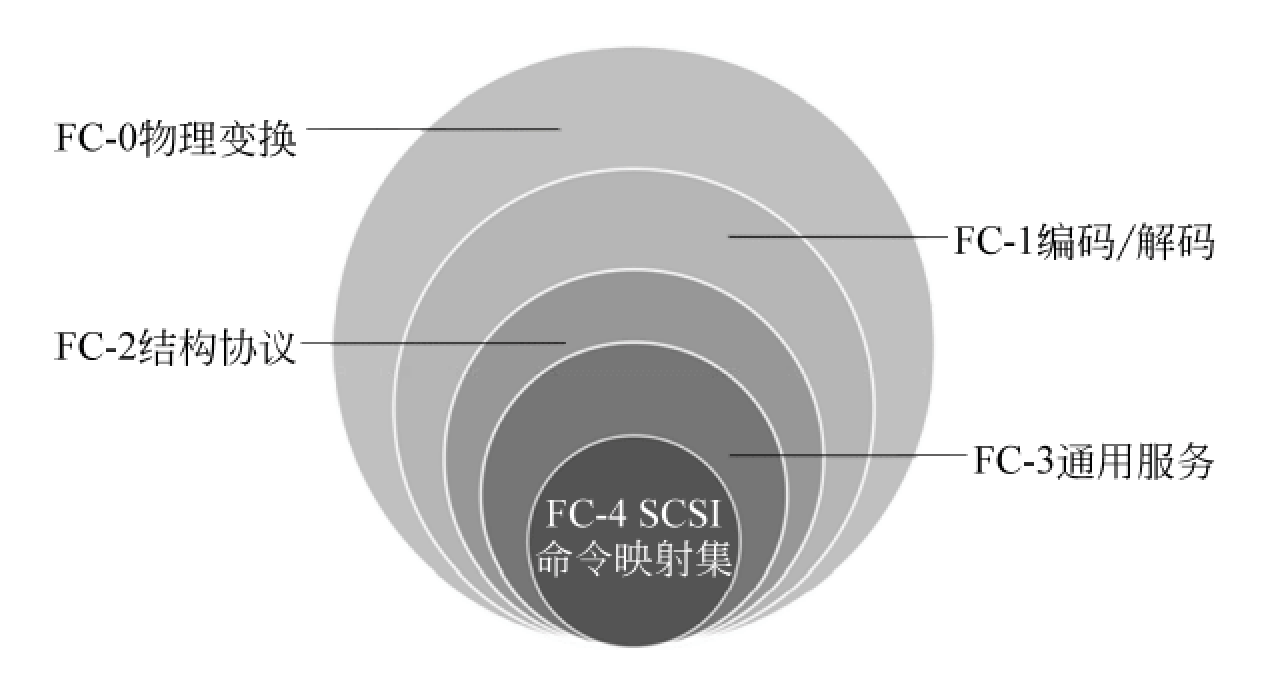


图 3-13　FC 与 SCSI 协议的关系

存储设备上的 FC 接口模块提供了应用服务器与存储系统的业务接口，用于接收应用服务器发出的数据交换命令，如图 3-14 所示。

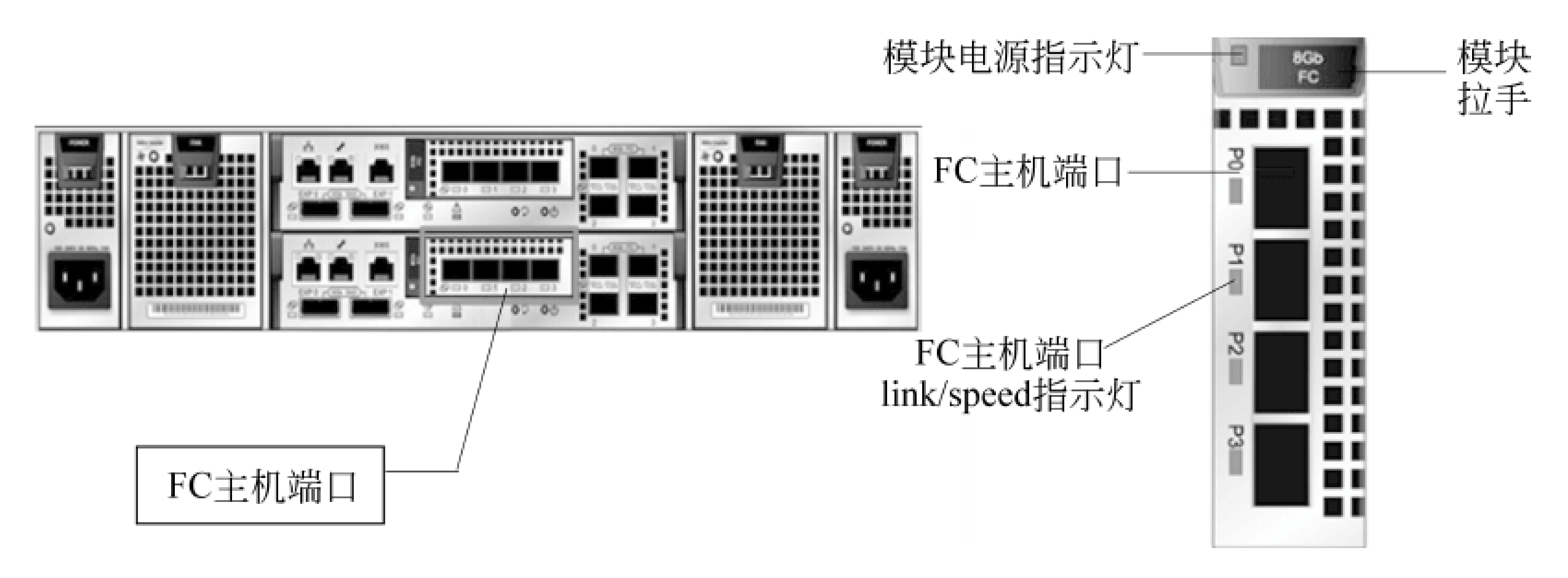


图 3-14　FC 存储设备

5．光纤交换机

光纤通道交换机使用光纤网络路由直接连接的方式，发起者和目标设备可以通过光纤通道交换机中的路由软件建立直接的连接以独享光纤的所有带宽。

光纤通道交换机是 SAN 的核心，它连接着主机和存储设备。一般可分为入门级交换机、工作组级光纤交换机、核心级光纤交换机。工作组级光纤交换机应用最多的领域是小型 SAN，通过级联交换机，能够建立一个大型的、虚拟的、具有分布式优点的交换机，并且它可以跨越的距离非常大。核心级交换机（又叫导向器）一般位于大型 SAN 的中心，支持光纤以外的协议、高级光纤服务（例如安全性、中继线和帧过滤等），核心光纤交换机往往采用基于“刀片式”的热插拔电路板。

与以太网交换机相比，FC 交换机用于构建光纤网络，而以太网交换机用于构建以太网络；光纤交换机中使用的是 FC 协议，以太网交换机使用的是 TCP/IP 协议。

光纤交换机上为了不同设备之间的访问隔离引入了 zone 的概念，zone 的功能类似于以太网交换机上的 VLAN 功能，它是将连接在 SAN 网络中的设备（主机和存储）在逻辑上划为不同的区域，使得不同区域中的设备相互间不能通过 FC 网络直接访问，从而实现网络中的设备之间的相互隔离。

6．FC 交换机端口

光纤网交换机中具有一些称为端口（Port）的连接部件，不同的端口根据其所连接的设

备类型的不同，所起到的作用是不同的。光纤通道标准定义了以下端口，如图3-15所示。

图3-15　FC交换机端口

(1) F_Port

F端口也被称为光纤网端口，用于将服务器和存储设备连接到交换机上。一个被连接到交换机F端口的设备就是一个节点(Node)，用光纤通道术语来说，它被看作是一个N端口(N_Port)。如果是在仲裁环路拓扑结构中，则被称为NL端口(NL_Port)。交换机通过特定的N端口或者是NL端口名称来识别这些光纤通道节点。

(2) E_Port

E端口也被称为扩展端口，用于进行交换机之间的连接。

(3) FL_Port

FC交换机的一个交换端口可以作为环路的组成部分，数据可以从交换机中传输到环上。在环路环境下正常工作的一个交换端口称之为FL_Port。

(4) G_Port

G端口是一种通用的端口，根据具体的实现方案，可以作为F端口或是E端口使用，也就意味着G端口实际上可以被用作两种端口功能的组合。由于G端口的自适应性，在进行FC-SAN的多交换机配置环境时，G端口为交换机提供了更好的灵活性并降低了每个端口所耗费的管理成本。

目前光纤交换机支持的端口速率有1Gb/s、2Gb/s、4Gb/s、8Gb/s。

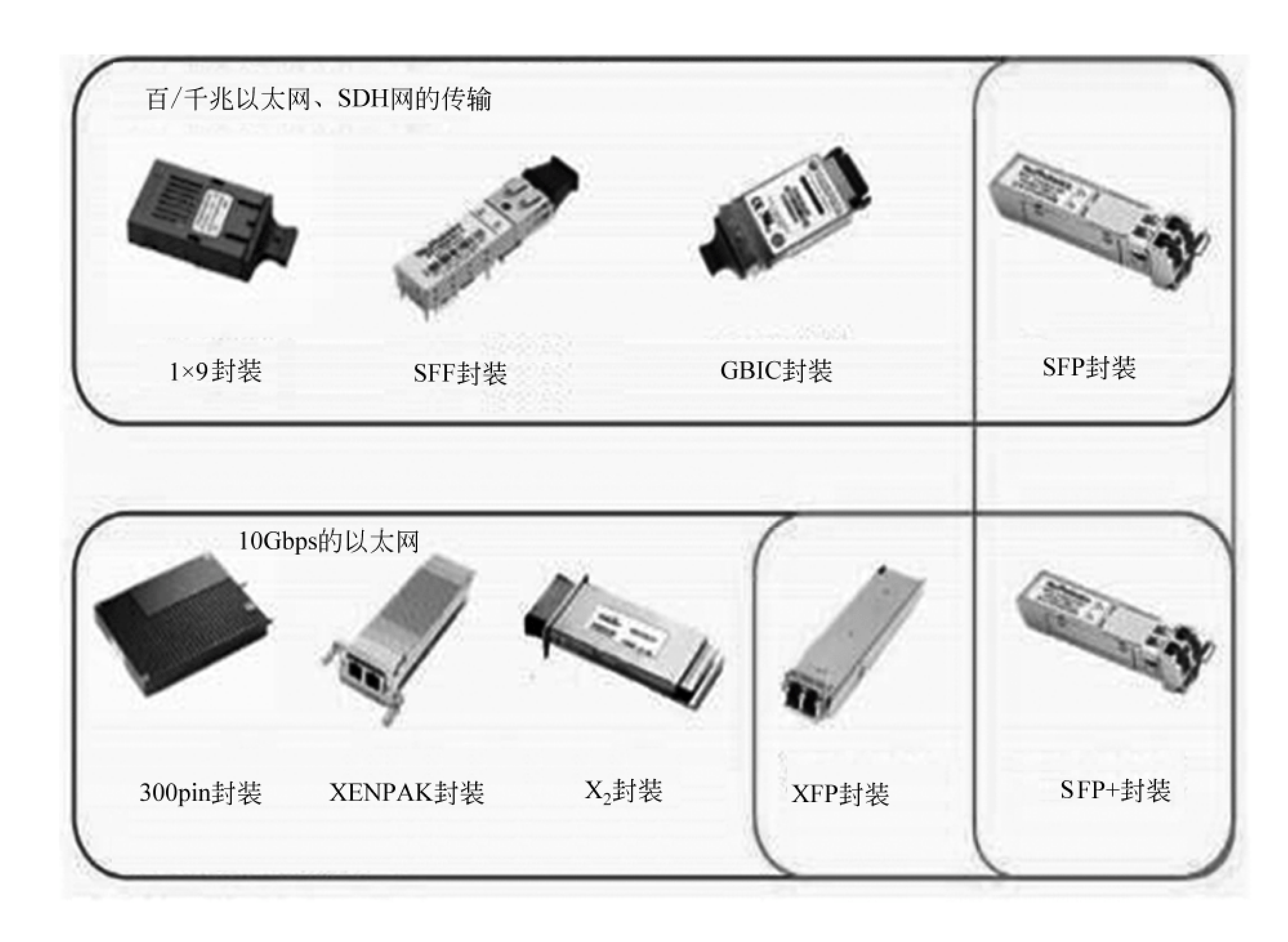
7. 常见的光模块

(1) 光通道交换机光模块由光电子器件、功能电路和光接口等组成。光电子器件包括发射和接收两部分，如图3-16所示。光模块可按不同类别划分。

① 按照速率分，包括以太网应用的100Base(百兆)、1000Base(千兆)，10GE SDH应用的155Mbps、622Mbps、2.5Gbps、10Gbps。

② 按照封装分，包括1×9、SFF、SFP、GBIC、XENPAK、XFP。

③ 按照光纤的类型分，包括单模光纤连接器、多模光纤连接器。



封装	简介	图片
1×9	1×9封装的光模块产品最早产生于1999年，SC接口，作为固定光模块使用	
GBIC	千兆以太网接口转换器，交换、路由产品广泛地采用GBIC模块。其可支持热插拔的特性，方便更新维护及故障定位	
SFF	SFF光模块是光模块产品演进的又一分支，目前广泛应用于EPON系统中的ONU侧	
SFP	小封装可插拔收发器，SFP光模块产品是最晚出现的光模块，也是目前应用最广泛的光模块产品。继承了GBIC的热插拔特性，也借鉴了SFF小型化的优势	
300Pin	最先被应用于SDH和10Gbps以太网光纤传输网络的模块，应用较少	
XENPAK	光模块产品演进中的重要一步，支持所有IEEE.802.3ae定义的光接口。技术成熟度较高，应用比较广泛，体积大，功耗大	
X_2	是Xenpak光模块的直接改进版，体积缩小了40%左右，成本高，只是作为一种过渡性的产品出现	
XFP	2002年提出的XFP多元协议，XFP光模块的出现和技术的飞速发展，以及其体积小，价格低的优势，得到广泛应用	
SFP+	具有比X_2和XFP封装更紧凑的外形尺寸，与SFP尺寸一样，成本比XFP产品低	

图 3-16 常见的光模块

④ 按照光纤连接器的连接头形式分，包括 FC、SC、ST、LC、MU、MTRJ 等，目前常用的有 FC、SC、ST、LC。

(2) 常用光纤连接介质如表 3-1 所示。

按照光纤的模式分类，可分为单模光纤(Single Mode Fiber)和多模光纤(Multi Mode Fiber)。

表 3-1　常用光纤连接介质

介质类型	发　射　器	速　率	距　离
9μm 单模光纤	1550nm 长波光激光器	1Gb/s	2m～50km
		2Gb/s	2m～50km
	1300nm 长波光激光器	1Gb/s	2m～10km
		2Gb/s	2m～2km
		4Gb/s	2m～2km
50μm 多模光纤	850nm 短波光激光器	1Gb/s	0.5～500m
		2Gb/s	0.5～300m
		4Gb/s	0.5～170m
62.5μm 多模光纤		1Gb/s	0.5～300m
		2Gb/s	0.5～150m
		4Gb/s	0.5～70m

8. HBA 卡

FC HBA 卡是联系服务器与存储区域网络的设备。与网络接口卡(Network Interface Card,NIC)类似,HBA 提供了服务器协议和光纤网络交换机之间进行转换的能力。HBA 连接到服务器的 PCI 总线,通过软件驱动程序来提供对光纤通道网络的支持。HBA 根据厂商的实现,可以使用单端口或者多端口配置。在多端口环境中,数据流拥有额外的数据路径,用于通过单个 HBA 在服务器和光纤网交换机之间传输数据。一个 HBA 可以拥有多个端口,而一个服务器也可以使用多个 HBA,这样的配置更具灵活性,可以实现到多个节点的单独连接,也能实现到同一节点的路径冗余以避免单点故障的风险,如图 3-17 所示。

HBA(Host Bus Adapter):主机总线适配器,就是连接主机I/O总线和计算机内存系统的I/O适配器。

分类:FC HBA、SCSI HBA、SAS HBA、iSCSI HBA等。

用途:用于服务器、海量存储子网络、外设间通过集线器、交换机和点对点连接进行双向、串行数据通信。

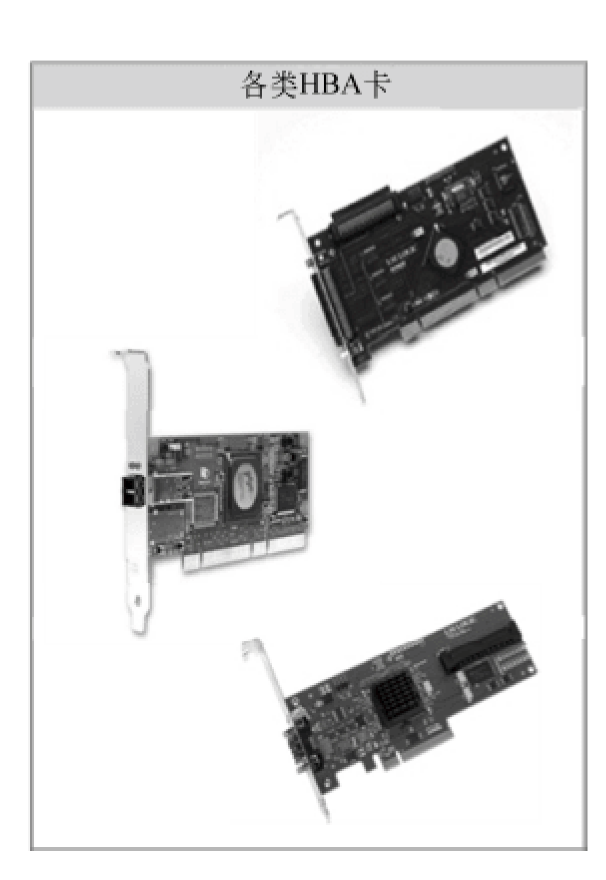


图 3-17　FC HBA 卡

FC HBA 的主要厂家有 Emulex、Qlogic、LSI、JNI(已经并入 AMCC,LSI 存储兼容列表写的 HBA 为 AMCC)、Agilent、Adaptec、IBM、HP、SUN。

9. FC HBA 卡的 WWN

FC HBA 的 WWN 具有两种类型。

(1) WWNN(World Wide Node Name):全球唯一节点名,由光纤通道通过映射分配给每一个上层节点一个全球唯一的 64 位标识符,一个 HBA 上的所有端口共享一个 WWNN。在计算机处理中,一个 WWNN 被分配给一个接入到光纤网中的节点(一个端点,如一个设备)。WWNN 可以被一个或者多个不同的端口(每个端口拥有不同的 WWPN,并且属于同一个节点)共同使用,如图 3-18 所示。

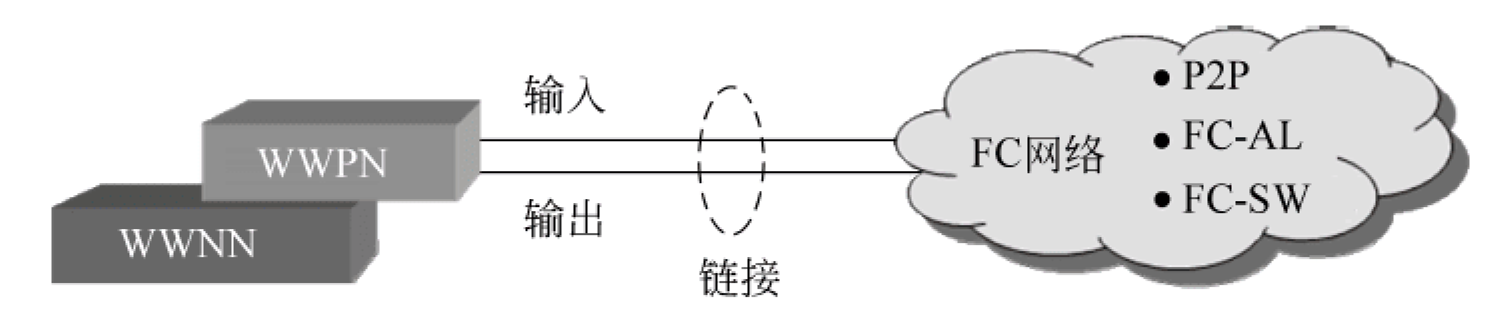


图 3-18 FC HBA 卡 WWN

(2) WWPN(World Wide Port Name):全球唯一端口名字,即分配给每一个光纤通道端口的全球唯一的 64 位标示符。每个 WWPN 被该端口独享。WWPN 在存储区域网络中的应用就等同于 MAC 地址在以太网协议中的应用。

3.3.3 SCSI 协议介绍

1. SCSI 协议与存储系统

计算机与存储系统之间的通信是通过总线来完成的。总线就是从源设备传输数据到目标设备的路径。在最简单的情况下,控制器的高速缓存作为源,将数据传输给目标磁盘。控制器首先向总线处理器发出请求使用总线的信号。该请求被接受之后,控制器高速缓存就开始执行发送操作。在这个过程中,控制器占用了总线,总线上所连接的其他设备都不能使用总线。当然,由于总线具备中断功能,所以总线处理器可以随时中断这一传输过程并将总线控制权交给其他设备,以便执行更高优先级的操作。

计算机中布满了总线——从一个位置向另一个位置传输信息和电力的高速通道。例如,将 MP3 或数码相机连接到计算机时,可能会使用通用串行总线(USB)端口。对于存储图片、音乐等的小型电子设备,USB 端口完全可以胜任传输数据和充电的工作。但是,这种总线还不足以同时支持整台计算机和服务器以及其他许多设备,如图 3-19 所示。

在这种情况下,就需要使用 SCSI 这样的总线。SCSI 直译为小型计算机系统专用接口(Small Computer System Interface),是一种连接主机和外围设备的接口,支持包括磁盘驱动器、磁带机、光驱、扫描仪在内的多种设备。它由 SCSI 控制器进行数据操作,SCSI 控制器相当于一块小型 CPU,有自己的命令集和缓存。SCSI 是一种特殊的总线结构,可以对计算机中的多个设备进行动态分工操作,对于系统同时要求的多个任务可以灵活机动地适当分配并动态完成。

SCSI 协议是主机与存储磁盘通信的基本协议。DAS 使用 SCSI 协议实现主机服务器与存储设备的互联。

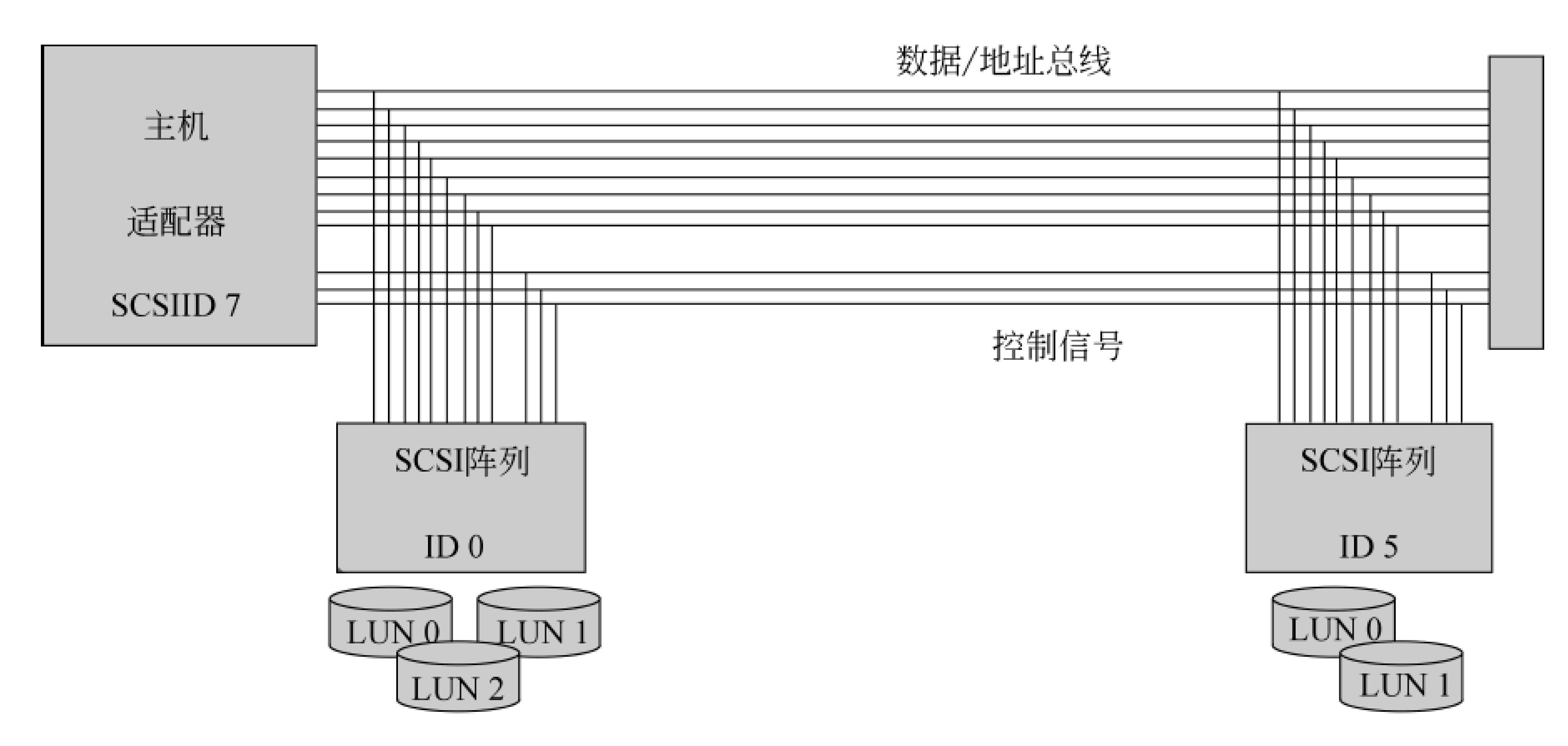


图 3-19　SCSI 总线

2. SCSI 协议模型

为了便于实现和理解 SCSI 的各个协议，SCSI 采取了分层结构。SCSI 可分为三层，即 SCSI 应用层、SCSI 传输层和 SCSI 互联层，如图 3-20 所示。

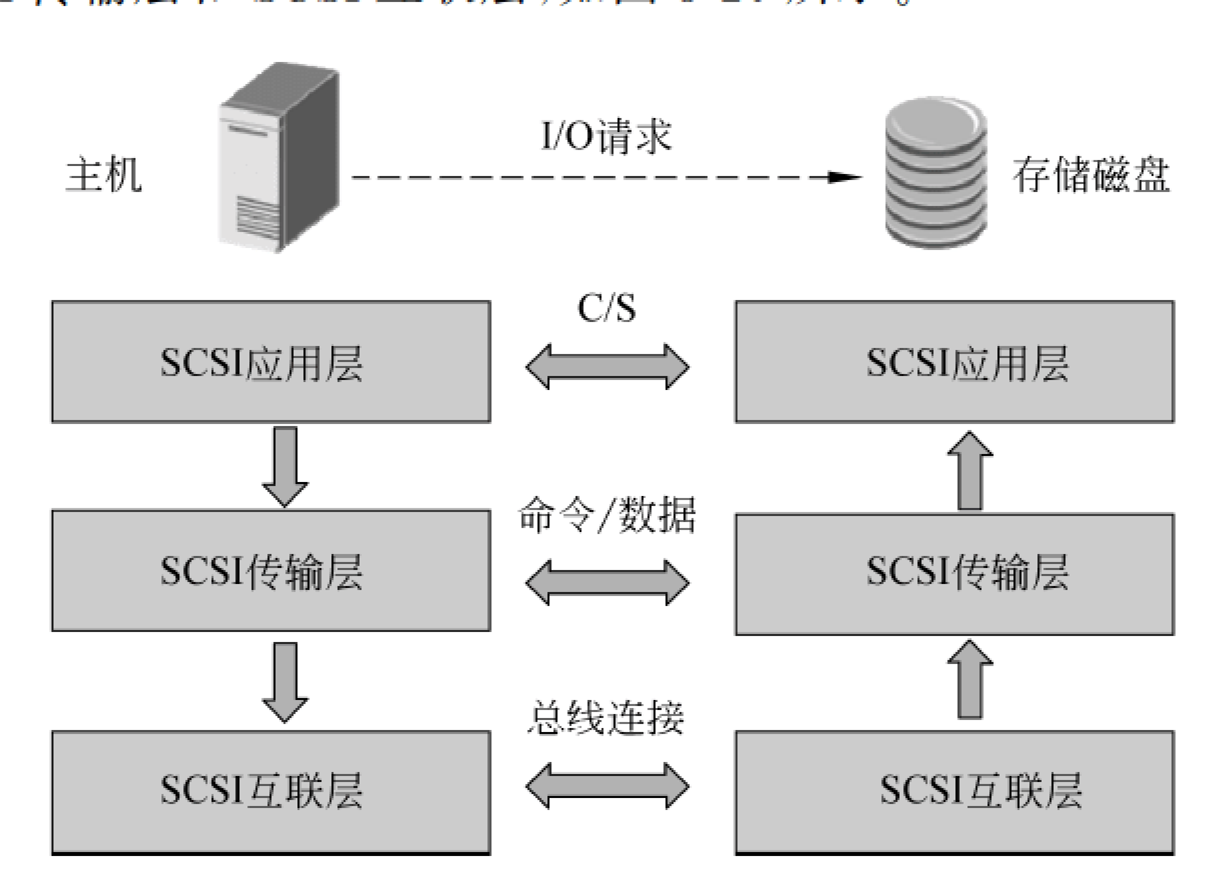


图 3-20　SCSI 协议模型

在应用层，SCSI 协议采用 C/S（客户/服务器）体系架构。SCSI 协议客户端位于主机中，代表上层应用程序、文件系统和操作系统发起 I/O 请求。SCSI 设备服务器位于目标设备（如存储磁盘）中，对客户端 I/O 请求做出响应。客户机/服务器请求和响应通过其下层协议进行传输。

在传输协议层，SCSI 设备之间通过一系列的命令实现数据块的传送，大致分成三个阶段：命令的执行，数据的传送和命令的确认。

SCSI 互联层完成 SCSI 设备对总线的连接以及发送方和目标方的选择等功能。

3. SCSI 协议寻址

为了对连接在 SCSI 总线上的设备寻址，SCSI 协议引入了 SCSI 设备 ID 和逻辑单元号 LUN。在 SCSI 总线上的每个设备都必须有一个唯一的 ID，其中包括服务器中的主机总线适配器也拥有设备 ID。每条总线最多可允许有 8 个或者 16 个设备 ID。

存储设备可能包括若干个子设备，如虚拟磁盘、磁带驱动器和介质更换器等。因此SCSI引入了逻辑单元号，以便于对存储设备中的子设备进行寻址。

传统的SCSI适配卡连接单个总线，相应地只具有一个总线号。一个服务器可能配置了多个SCSI控制器，从而就可能有多条SCSI总线。在引入存储网络之后，每个光纤通道HBA(Host Bus Adapter)或iSCSI(Internet SCSI)网卡也都连接一条总线，必须对每一条总线分配一个总线号，在它们之间依靠不同的总线号加以区分。因此，可以使用一个三元描述标识一个SCSI目标：总线号、设备ID、逻辑单元号，如图3-21所示。

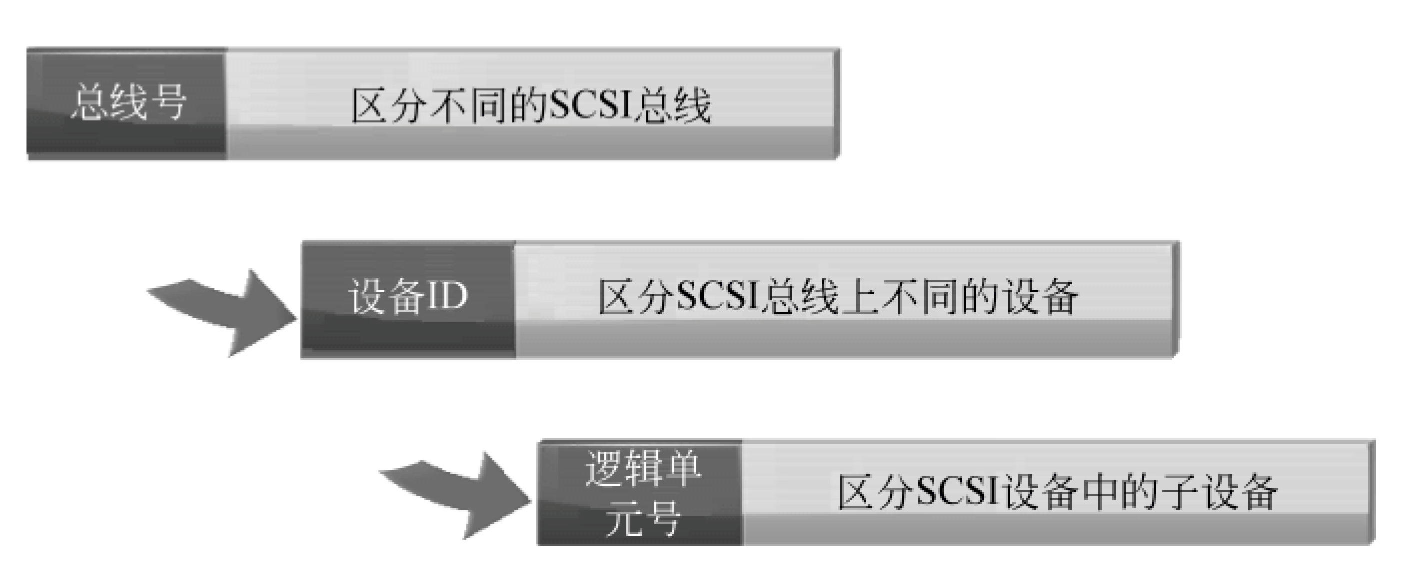


图3-21　SCSI协议寻址

4. Windows系统中查看SCSI设备ID的方式

在Windows桌面中右击“我的电脑”，选择“管理”，再选择“磁盘管理”→“映射的磁盘”，右击并选择“属性”，在“常规”选项卡中查看“位置”选项的内容，就是SCSI ID信息，如图3-22所示。



图3-22　Windows系统中查看SCSI设备ID

5. Linux 系统中查看 SCSI 设备 ID 的方式

图中每一行前面方括号中条目分别是 SCSI host、channel、target number、LUN tuple，每个元素由冒号分开。当有多个 SCSI 设备条目时，按元组升序排列。

在 Linux 系统的命令行中输入命令 lsscsi，如图 3-23 所示。

```
linux-suse-icy:/proc/scsi # lsscsi
[0:0:0:0]    disk    ATA      ST3160318AS        CC65  /dev/sda
[0:0:1:0]    cd/dvd  HL-DT-ST DVD-ROM DH10N      0M10  /dev/sr0
[2:0:0:0]    disk    HUAWEI   S5500T             2105  /dev/sdb
```

图 3-23　Linux 系统查看 SCSI 设备 ID

6. SCSI 命令描述块（CDB）

在互联层完成 SCSI 设备对总线的连接，以及发送方和目标方选择的基础上，传输层协议执行实际的数据传输。传输协议的运行过程包括发送命令、传输数据和对命令执行的确认。发起方通过命令描述块（CDB）向目标方发送具体的命令。命令描述块有定长和不定长两种格式，而定长格式又有 6、10、12、16 字节不同的长度规定。

（1）操作码

操作码是所有命令描述块都有的，它总是被放在命令描述块的开头一个字节。5～7 位是组代码，指示该命令具体属于哪个命令组，它决定 CDB 的长度；0～4 位则是具体的命令代码。8 比特在理论上共有 256 个可能的操作码。

（2）命令参数

① 混杂 CDB 信息：该参数表示与具体的 CDB 相关的信息，如表示逻辑设备号。

② 逻辑块地址：该地址是逻辑单元（比如磁盘）中的起始操作块的位置。

③ 传送长度：该长度表示命令所请求的传送量，通常是块数。

④ 参数表长度：表示需要传送到存储设备的参数的长度，0 表示不需要传递参数。

⑤ 分配长度：分配长度表示应用客户为缓冲区分配的最大长度，根据具体的 CDB 类别，可能是字节数，也可能是块数。

发起方通过命令描述块（Command Description Block）向目标方发送具体的命令。

SPC（SCSI Primary Commands，SCSI 基础命令）规范定义了 CDB 的标准。CDB 结构如图 3-24 所示。

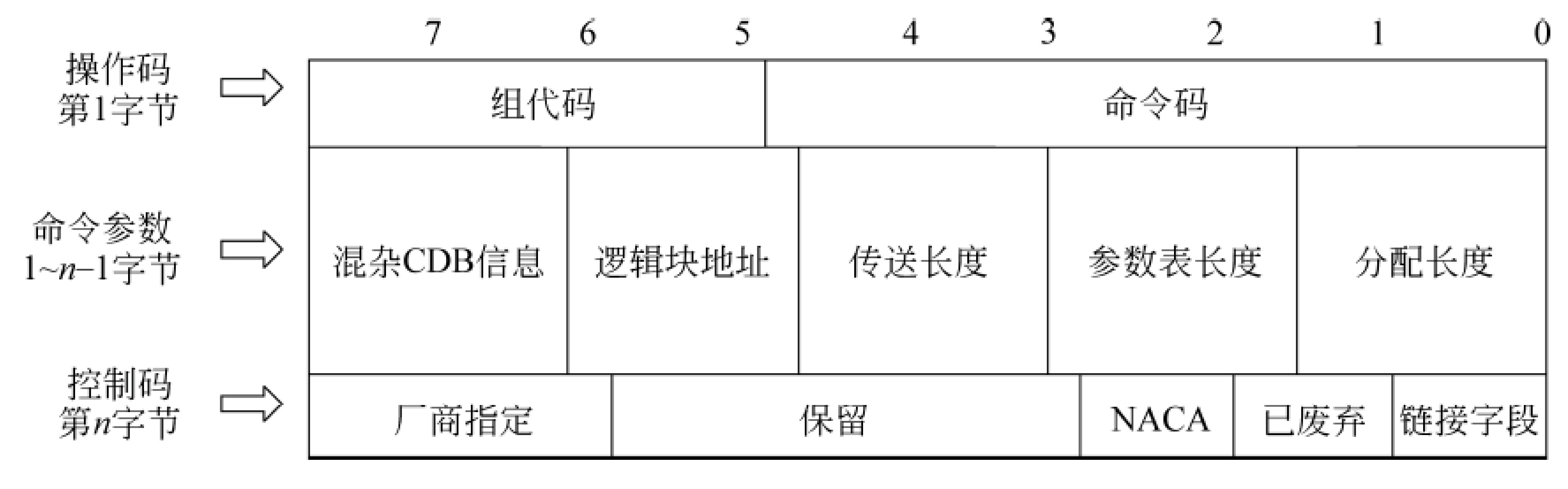


图 3-24　SCSI 命令描述块（CDB）

（3）控制码

所有 CDB 格式的最后一个字节为控制码。

NACA 比特是为了让应用能够事先声明哪些命令执行的错误或异常需要善后处理，指

定当指令返回 CHECK CONDITION 状态的时候，自动应急处理(Auto Contingent Allegiance，ACA)是否会被创建。

链接比特可以被用作跨越多个指令延续任务。比特为 1 时表示发起方请求跨越多个 SCSI 指令延续任务。

发起方通过命令描述块(command description block)向目标方发送具体的命令。

7. SCSI 读/写操作过程

主机需要从存储设备获取数据，SCSI 读操作需要完成以下步骤。

(1) 主机操作系统需首先将用户的读取操作通过 SCSI I/O 的应用程序编程接口(Application Programming Interface，API)转化为 SCSI 的读操作，并在操作完成后通过相应的 API 返回响应的值。

(2) 发起方 SCSI 总线由空闲阶段进入总线仲裁和选择阶段，获得对总线使用权以及对目标方的选择和寻址。

(3) 发起方通过 CDB 向目标方发送 SCSI 的读命令。

(4) 目标方接收到该命令，通过设备管理器在指定的逻辑单元中执行该命令请求的操作。

(5) 目标方以字节为单位向发起方传送所需要的数据。

(6) 在数据传输完毕后，目标方向发起方发送命令完成的报告。

(7) 发起方接收到命令完成的响应，总线可恢复到空闲阶段。

SCSI 的写操作过程与读操作过程类似，但数据传送的方向不同，它把数据从发送方向目标方传送，如图 3-25 所示。

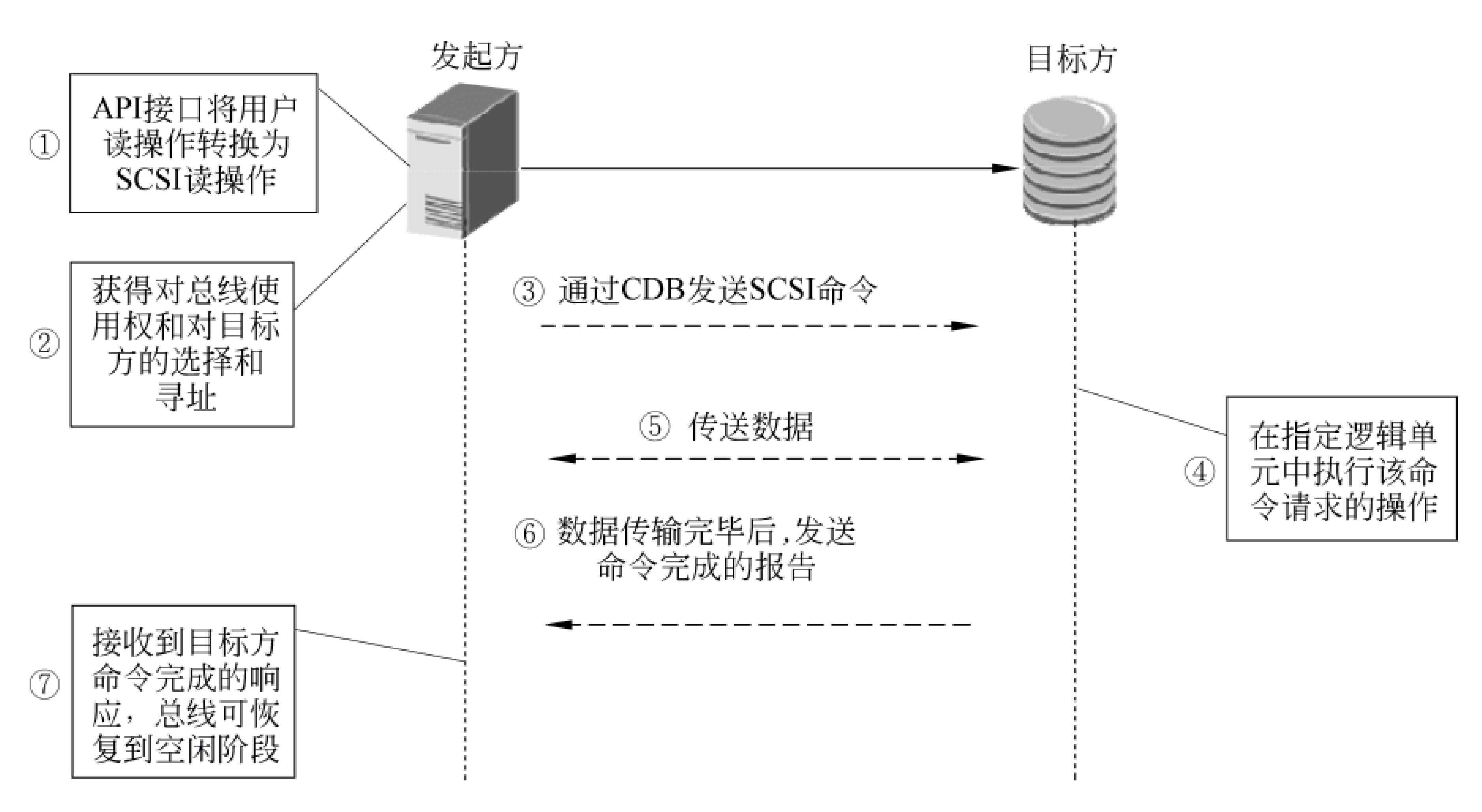


图 3-25 SCSI 读/写操作过程

8. SCSI 协议的常见类型

目前常见的 SCSI 类型及各自性能特征如表 3-2 所示。

(1) SCSI-1：它是最早的 SCSI 接口，它的特点是支持同步和异步 SCSI 外围设备，支持 7 台 8 位的外围设备，最大数据传输率为 5MB/s。

表 3-2　SCSI 类型及各自性能特征

接口模式	传输率(MB/s)	数据频宽(bits)	可连接设备数	接口模式	传输率(MB/s)	数据频宽(bits)	可连接设备数
SCSI-1	5	8	8	Ultra 2 SCSI	80	16	16
SCSI-2	10	8	8	Ultra-160 SCSI	160	16	16
SCSI-3(Ultra SCSI)	20	8	8	Ultra-320 SCSI	320	16	16
SCSI-3(Ultra Wide SCSI)	40	16	16	Ultra-640 SCSI	640	16	16

(2) SCSI-2：它是 SCSI-1 的后续接口，是 1992 年提出，也称为 Fast SCSI。如果采用原来的 8 位并行数据传输则称为 Fast SCSI，它的数据传输率为 10MB/s，最大支持连接设备数为 7 台。

(3) SCSI-3：它是在 SCSI-2 之后推出的。如果采用原来的 8 位并行数据传输时称为 Ultra SCSI，它的数据传输率为 20MB/s，最大支持连接设备数为 7 台。在将并行数据传输的总线带宽提高到 16 位后出现了 Ultra Wide SCSI，它的传输率又成倍提高，即达到了 40MB/s，最大支持连接设备数为 15 台。

(4) Ultra2 SCSI：它是在 Ultra SCSI 的基础上推出的 SCSI 接口类型。

(5) Ultra160 SCSI：它是 Ultra2 SCSI 的更新接口，使用和 Ultra2 SCSI 完全一样的接口电缆及终结器，但是由于 Ultra160 SCSI 采用双缘传输频率(Double Transition Clocking)，而 Ultra2 SCSI 采用的是单缘传输频率，因此 Ultra160 SCSI 的传输率是前者的两倍，即 160MB/s。

(6) Ultra320 SCSI：它的技术规范为 SCSI-3 或 SPI-4。Ultra320 SCSI 单通道的数据传输速率最大可达 320MB/s。

(7) Ultra640 SCSI：它的技术规范为 SCSI-3 或 SPI-5。Ultra640 SCSI 的数据传输速率最大可达 640MB/s。

3.3.4　iSCSI 协议介绍

1. iSCSI 协议

通过 SCSI 控制卡的使用可以连接多个设备，形成自己的“网络”，但是这个“网络”仅局限于与所附加的主机进行通信，并不能在以太网上共享。如果能够通过 SCSI 协议组成网络，并且能够直接挂载到以太网上作为网络节点和其他设备进行互联共享，那么 SCSI 就可以得到更广泛的应用。所以，经过对 SCSI 的改进，就推出了 iSCSI 这个协议。基于 iSCSI 协议的 IP-SAN 是把用户的请求转换成 SCSI 代码，并将数据封装进 IP 包内并在以太网中传输。

iSCSI 方案最早是由 Cisco 和 IBM 两家发起，并且由 Adaptec、Cisco、HP、IBM、Quantum 等公司共同倡导，它提供基于 TCP 传输并将数据驻留在 SCSI 设备上的方法。iSCSI 标准草案在 2001 年推出，并经过多次论证和修改，于 2002 年提交 IETF，在 2003 年 2 月，iSCSI 标准正式发布。iSCSI 技术的重要贡献在于其对传统技术的继承和发展：其一，SCSI(Small Computer System Interface，小型计算机系统接口)技术是被磁盘、磁带等设备广泛采用的存储标准，从 1986 年诞生起到现在仍然保持着良好的发展势头；其二，沿用

TCP/IP 协议,TCP/IP 在网络方面是最通用、最成熟的协议,且 IP 网络的基础建设非常完善。这两点为 iSCSI 的无限扩展提供了坚实的基础。

IP 网络的普及性将使得数据可以通过 LAN、WAN 或者是通过 Internet 利用新型 IP 存储协议传输,iSCSI 即是在这个思想的指导下进行研究和开发的。iSCSI 是基于 IP 协议的技术标准,实现了 SCSI 和 TCP/IP 协议的融合,对众多的以太网用户而言,只需要极少的投资,就可以方便、快捷地对信息和数据进行交互式传输和管理。

iSCSI(Internet SCSI)把 SCSI 命令和块状数据封装在 TCP 中在 IP 网络传输,基本出发点是利用成熟的 IP 网络技术来实现和延伸 SAN,如图 3-26 所示。



图 3-26　IP 网络技术 SAN

2. iSCSI 体系结构

在支持 iSCSI 的系统中,用户在一台 SCSI 存储设备上发出存数据或取数据的命令,操作系统对该请求进行处理,并将该请求转换成一条或者多条 SCSI 指令,然后再传给目标 SCSI 控制卡。iSCSI 节点将指令和数据封装(Encapsulation)起来,形成一个 iSCSI 包,然后该数据封装被传送给 TCP/IP 层,再由 TCP/IP 协议将 iSCSI 包封装成 IP 协议数据以适合在网络中传输。也可以对封装的 SCSI 命令进行加密处理,然后在不安全的网络上传送。

数据包可以在局域网或 Internet 上传送。在接收存储控制器上,数据报重新被组合,然后存储控制器读取 iSCSI 包中的 SCSI 控制命令和数据并发送到相应的磁盘驱动器上,磁盘驱动器再执行初始计算机或应用所需求的功能。如果发送的是数据请求,那么将数据从磁盘驱动器中取出并进行封装后,发送给发出请求的计算机,而整个过程对于用户来说都是透明的。尽管 SCSI 命令的执行和数据准备可以通过使用标准 TCP/IP 和现成的网络控制卡的软件来完成,但是在利用软件完成封装和解封装的情况下,在主机处理器上实现这些功能需要很多的 CPU 运算周期来处理数据和 SCSI 命令。如果将这些事务交给专门的设备处理,则可以将对系统性能的影响减少到最小程度,因此,发展在 iSCSI 标准下并执行 SCSI 命令和完成数据准备的专用 iSCSI 适配器是有必要的。iSCSI 适配器结合了 NIC 和 HBA 的功能。这种适配器以块方式取得数据,利用 TCP/IP 处理引擎在适配卡上完成数据分化和

处理，然后通过 IP 网络送出 IP 数据包。这些功能的完成使用户可以在不降低服务器性能的基础上创建一个基于 IP 的 SAN。

iSCSI 节点将 SCSI 指令和数据封装成 iSCSI 包，然后该数据封装被传送给 TCP/IP 层，再由 TCP/IP 协议将 iSCSI 包封装成 IP 协议数据以适合在网络中传输，如图 3-27 所示。

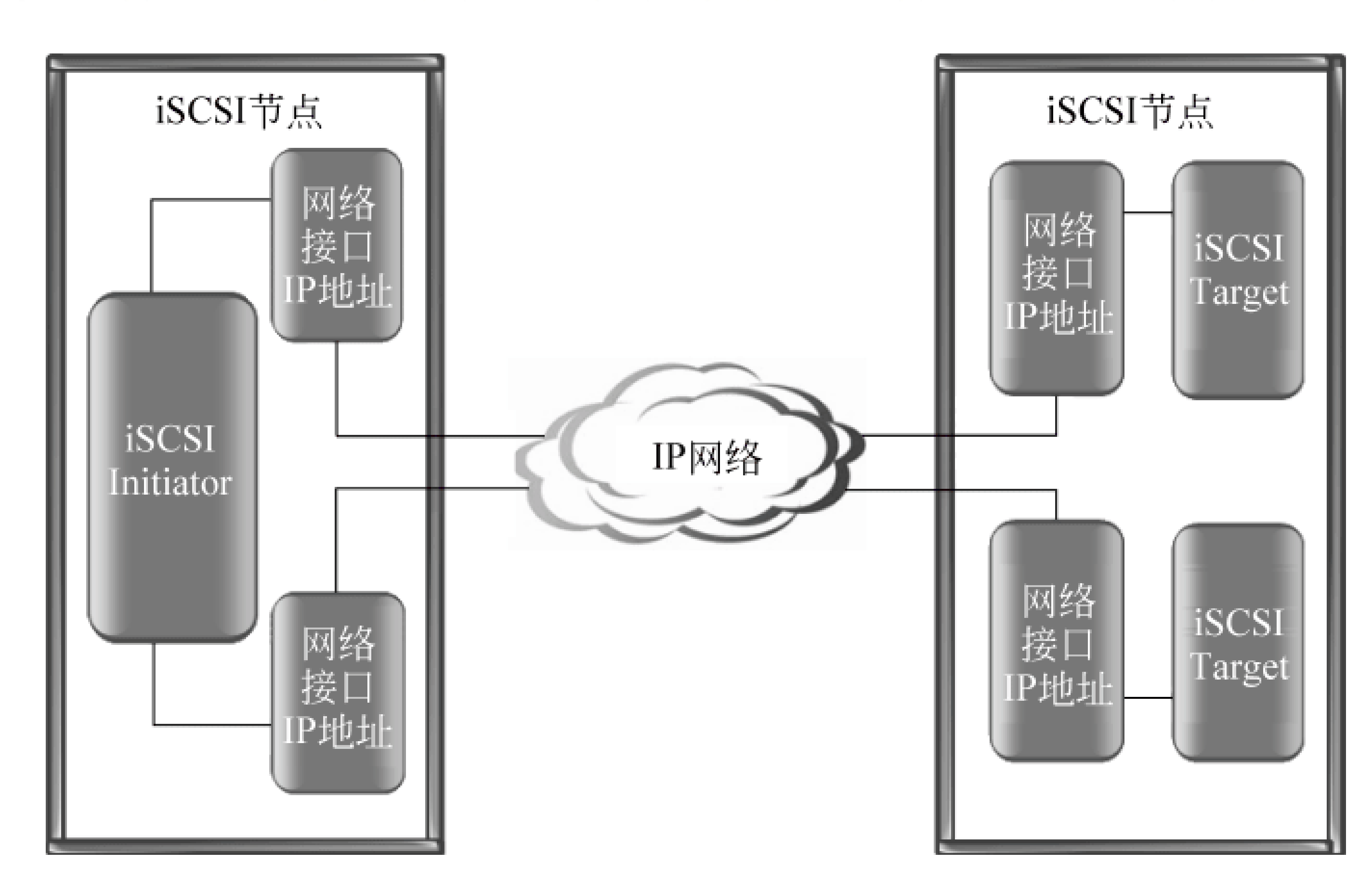


图 3-27　iSCSI 体系结构

3. iSCSI 的发起端与目标端

iSCSI 的通信体系仍然继承了 SCSI 的部分特性，在 iSCSI 通信中，具有一个发起 I/O 请求的启动设备（Initiator），以及一个响应请求并执行实际 I/O 操作的目标设备（Target）。在 Initiator 和 Target 建立连接后，Target 在操作中作为主设备控制整个工作过程。

（1）iSCSI Initiator

iSCS Initiaor 可分为三种，即软件 Initiator 驱动程序、硬件的 TOE（TCP Offload Engine，TCP 卸载引擎）卡以及 iSCSI HBA 卡。就性能而言，软件 Initiator 驱动程序最差，TOE 卡居中，iSCSI HBA 卡最佳。

（2）iSCSI Target

iSCSI Target 通常为 iSCSI 磁盘阵列、iSCSI 磁带库等。

iSCSI 协议为 Initiator 和 Target 定义了一套命名和寻址方法。所有的 iSCSI 节点都是通过其 iSCSI 名称被标识的。这种命名方式使得 iSCSI 名称不会与主机名混淆。

iSCSI 使用 iSCSI Name 来唯一鉴别启动设备和目标设备。地址会随着启动设备和目标设备的移动而改变，但是名字始终是不变的。建立连接时，启动设备发出一个请求，目标设备接收到请求后，确认启动设备发起的请求中所携带的 iSCSI Name 是否与目标设备绑定的 iSCSI Name 一致，如果一致，便建立通信连接。每个 iSCSI 节点只允许有一个 iSCSI Name，一个 iSCSI Name 可以被用来建立一个启动设备到多个目标设备的连接，多个 iSCSI Name 可以被用来建立一个目标设备到多个启动设备的连接，如图 3-28 所示。

4. iSCSI 数据包封装模型

支持 iSCSI 的服务器可以配置一块专用的 iSCSI 主机总线适配器卡。所有的 SCSI 命令都被封装成 iSCSI 协议数据单元（Protocol Data Unit，PDU），iSCSI 会利用 TCP/IP 协议栈中传输层的 TCP 协议为连接提供可靠的传输机制，在封装 TCP 数据段头以及 IP 数据包

启动器(Initiator)

- SCSI层负责生成CDB(命令描述符块),将CDB传给iSCSI
- iSCSI层负责生成iSCSI PDU(协议数据单元),并通过IP网络将PDU发给Target

目标器(Target)

- iSCSI层收到PDU,将CDB传给SCSI层
- SCSI层负责解释CDB的意义,必要时发送响应

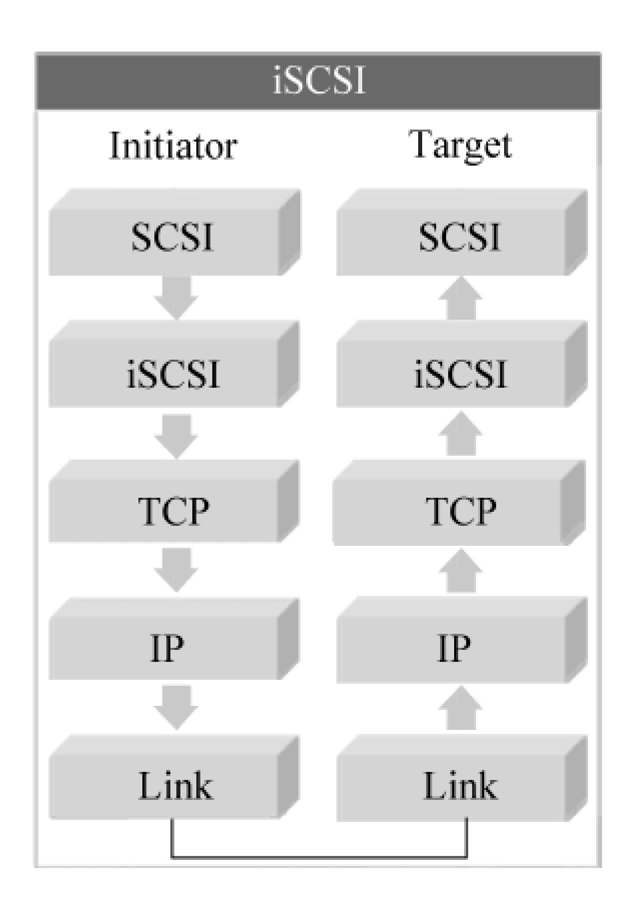


图 3-28 iSCSI 的启动端与目标端

头后,其内部所封装的 SCSI 命令或数据对于底层网络设备而言是不可见的,网络设备只会将其视为普通 IP 数据包进行传递,从而实现了 SCSI 指令和数据的透明传输。

iSCSI 协议是将 SCSI 的远程过程调用(Remote Procedure Call,RPC)映射到 IP 协议的过程。iSCSI 协议提供了独立于其所携带的 SCSI CDB 层的概念。iSCSI 请求传递 SCSI 命令,iSCSI 响应处理 SCSI 响应和状态。iSCSI 为基于 IP 协议的 PDU 提供了一个在 SCSI 的命令结构内映射的机制,SCSI 的命令及参数被填充在一定长度的数据块内进行传输。一个 iSCSI 翻译器取得 SCSI CDB(Command description Block,命令描述块),并将其映射为 iSCSI PDU,在 TCP 连接上发送到一个目标 iSCSI 设备。翻译器通过连接 ID 识别一组映射 SCSI 连接的 TCP 连接。从启动设备和目标设备的角度来看,这个连接就像是一个普通的 SCSI 通信一样,整个 IP 传输对于启动器设备和目标设备而言是透明的。启动设备或目标设备可以是一个 iSCSI 设备,能够用 TCP 直接在 IP 网中通信。

所有的 SCSI 命令都被封装成 iSCSI 协议数据单元,iSCSI 利用 TCP/IP 协议栈中传输层的 TCP 协议为连接提供可靠的传输机制,如图 3-29 所示。

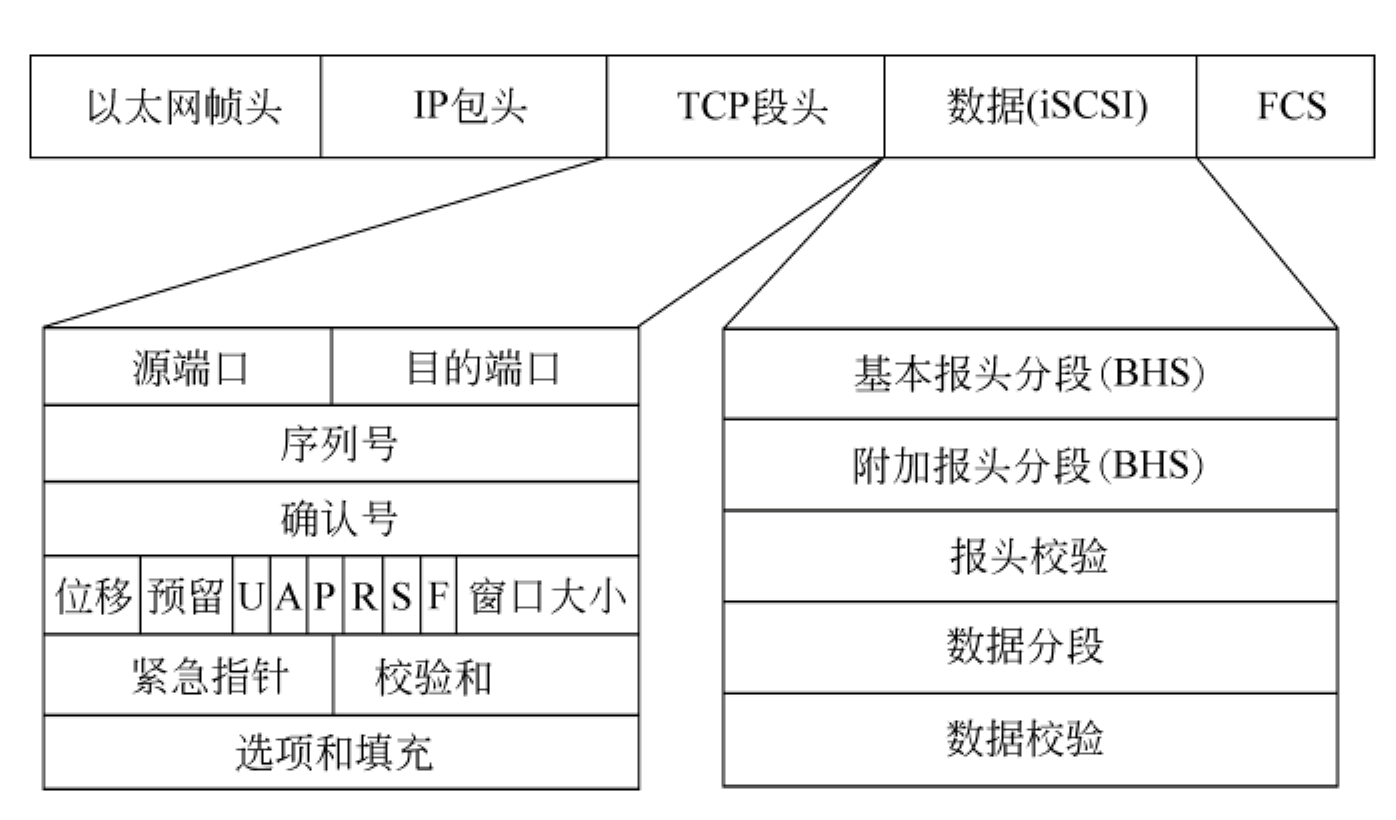


图 3-29 iSCSI 数据包封装模型

3.3.5　IP SAN 的产生与发展

1. FC SAN 与 IP SAN

前面介绍了基于 FC 协议的 FC SAN，主要应用于对于性能、冗余度和可获得性都有较高要求的中高端存储需求。由于其高昂的价格、技术和配置复杂、有限的传输距离、不同厂商设备互联共享等问题，也限制了其使用的范围。为了提高 SAN 的普及程度和应用范围，并充分利用 SAN 本身所具备的架构优势，SAN 的发展方向开始考虑和已经普及并且相对廉价的 IP 网络进行融合。

基于 TCP/IP 协议的以太网的 IP-SAN 存储开始进入人们的视野，并得到了网络厂商的广泛支持。与基于光纤通道技术的 FC-SAN 相比较而言，IP-SAN 存储系统具有节约大量成本、加快实施速度、优化可靠性以及增强扩展能力等优点。

简单而言，IP SAN 存储就是使用 IP 网络而不使用光纤网络来做服务器和存储设备的连接技术。IP SAN 存储是基于 IP 网络来实现块级数据存储的方式。目前除了标准已获通过的 iSCSI，还有 FCIP、iFCP 等标准。其中 iSCSI 发展最为迅速，已经成为 IP 存储的中流砥柱。基于 iSCSI 的 SAN 的目的就是要使用本地 iSCSI Initiator（启动器，通常为服务器），通过 IP 网络和 iSCSI Target（目标器，通常为存储设备）来建立 SAN 网络，如图 3-30 所示。

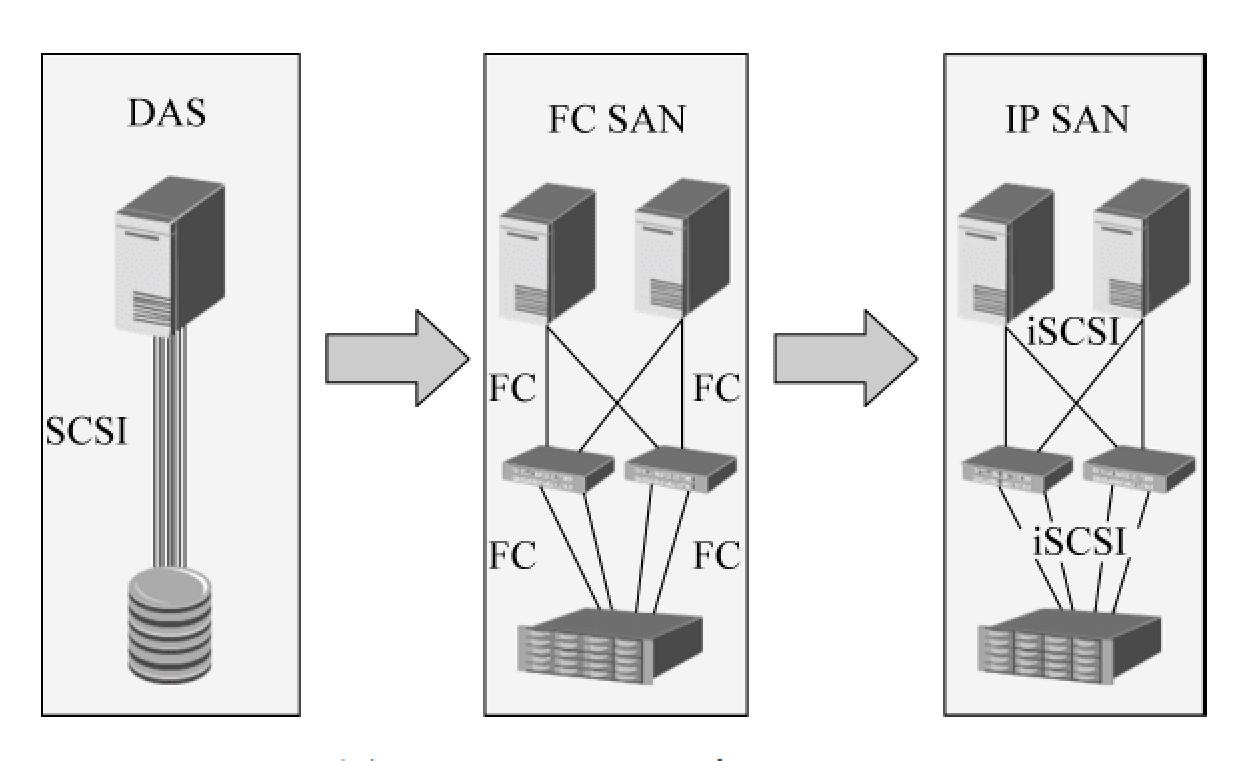


图 3-30　iFC SAN 与 IP SAN

2. IP SAN

IP SAN 是标准的 TCP/IP 协议和 SCSI 指令集相结合的产物，以其协议标准化、整体成本低廉和维护简便等优势成为网络存储领域的重要产品形态。

IP SAN 是基于 IP 网络来实现数据块传输的网络存储形态，与传统 FC SAN 的最大区别在于传输协议和传输介质的不同。目前常见的 IP SAN 协议有 iSCSI、FCIP、iFCP 等，其中 iSCSI 是发展最快的协议标准，人们所说的 IP SAN 大多数就是指基于 iSCSI 实现的 SAN。

IP SAN 把 SCSI 指令集封装在了 TCP/IP 上。这就好比不管我们是选择哪家快递公司，最终都是把我们想要发送的东西发送至目的地，都是由我们发起寄送请求，快递公司进行响应，差别只在于快递公司不同而已。iSCSI 则是全新建立在 TCP/IP 和 SCSI 指令集的基础上的标准协议，所以其开放性和扩展性更好。这也是其大行其道的原因。

以 TCP/IP 协议为底层传输协议，采用以太网作为承载介质构建起来的存储区域网络架构，实现 IP-SAN 的典型协议是 iSCSI，它定义了 SCSI 指令集在 IP 中传输的封装方式，如图 3-31 所示。

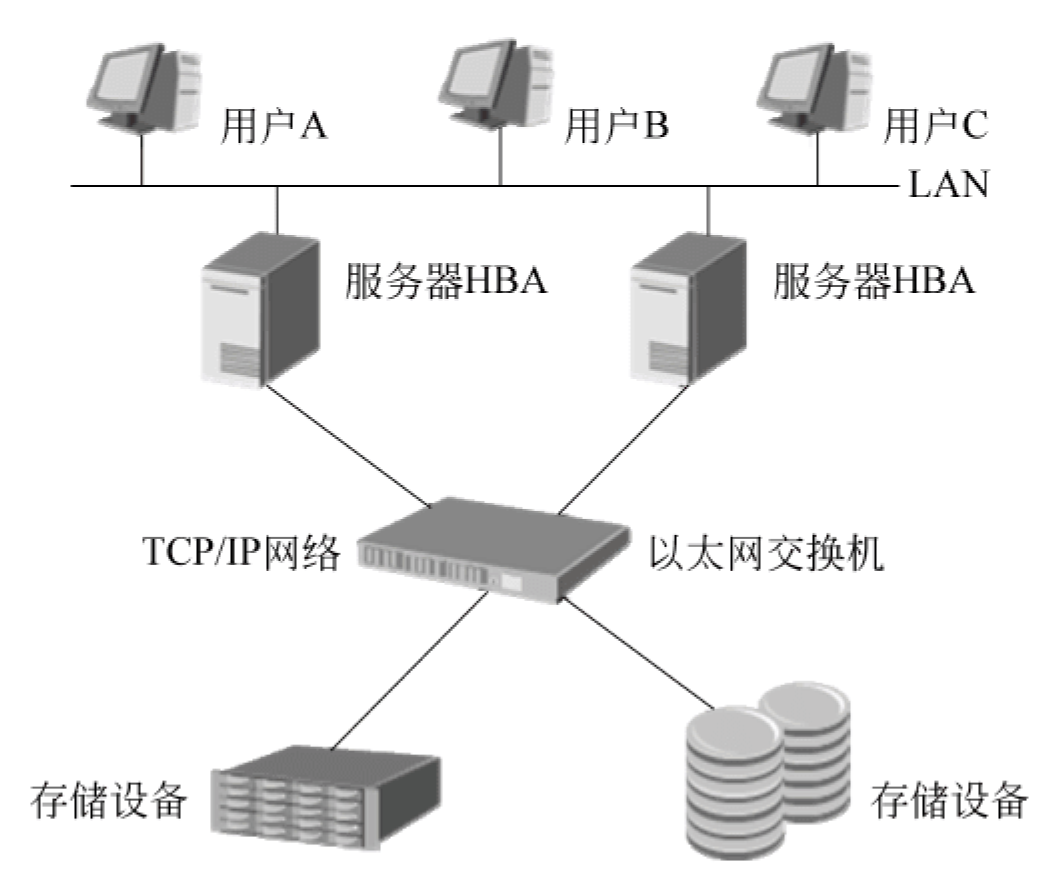


图 3-31　IP-SAN 的典型协议 iSCSI

3. IP SAN 的优势

IP SAN 主要基于 iSCSI 实现。iSCSI 协议是建立在 TCP/IP 协议和 SCSI 指令集的基础之上的标准化协议。正是其优良的基因决定了其巨大的优势。

那么，TCP/IP 和 SCSI 有什么优势呢？

第一，它们都是标准化协议，所以有大量的标准化设备可供采用。

第二，它们都是已经发展多年的成熟性协议，具有广泛的群众基础。

第三，作为标准在各类软件硬件开发中已经广泛采用。

IP SAN 全盘继承了父母的优良基因，从而具备了很多方面的优势。通过这些优势，给客户带来了哪些好处呢？最重要的一点就是 IP SAN 总体拥有成本（TCO）低，非常有利于其广泛地应用和推广。总体拥有成本（TCO）是包含很多内容的，比如建设一个存储系统，则需要购买磁盘阵列、接入设备（HBA 和交换机），需要人员培训、日常维护、后续扩容、容灾扩展等。IP SAN 因为 IP 网络的广泛应用优势，可以大幅降低单次采购的接入设备采购成本、减少维护成本，而且后续扩容和网络扩展成本也大幅降低，如图 3-32 所示。

4. IP SAN 面临的挑战

尽管 IP 存储标准早已建立且应用，但是将其真正广泛应用到存储环境中还需要解决以下几个关键问题。

（1）数据安全性：企业网络中最重要的还是数据，所以，SAN 中保存的数据的安全性和可靠性应当受到格外的重视。传统的 FC-SAN 由于 FC 网络的异构性，和传统的业务 IP 网络从物理上隔离，从而保证了在 SAN 中传输和存储的数据安全性。然而，当存储设备通过 IP 架构进行远程连接时，尽管 IP 协议可以应用 IPSec 以保障数据的安全性，但也只能提供数据在网络传输过程的动态安全性，并不能保证数据被保存在存储设备上的静态安全性。由于 IP 网络是开放式网络，仍然存在众多安全漏洞，并且使用 IP 网络构建的 IP-SAN 和传统的 IP 业务很难从物理上完全隔离。所以，提高数据在传输过程的安全性和在存储设备中



图 3-32　IP-SAN 的优势

的安全性是 IP 存储面临的严峻问题。

（2）TCP 负载空闲引擎：由于 IP 协议是无连接不可靠的传输协议，数据的可靠性和完整性是由 TCP 协议来提供的。而 TCP 为了完成数据的排序工作，需要占用较多的主机 CPU 资源，从而导致用户业务处理延迟的增加。所以，iSCSI 可以采用一种被称为 TCP 卸载引擎（TCP Off－loading Engine，TOE）的设备，将原本需要由 CPU 处理完成的 TCP 封装和解封装过程下移至 TOE 网卡完成，使 CPU 更专注于用户事务请求处理和数据包本身的处理，可以有效地降低主机 CPU 的负载，从而提升 CPU 的处理性能。

（3）块数据传输问题：FC 存储协议具有速率高、延迟低的特点，适合传输大块的数据（Block Data）；而从网络协议上来看，IP 协议传输速率相对较低、延迟较高，比较适合传输大量的小块消息。并且，FC 在传输数据时将数据封装为 2KB 左右的数据帧进行传输，而以太网则将数据封装为 1.5KB 的数据包在 IP 网络中进行传递，所以 IP 协议对大块数据的传输的效率还有待提高，如图 3-33 所示。

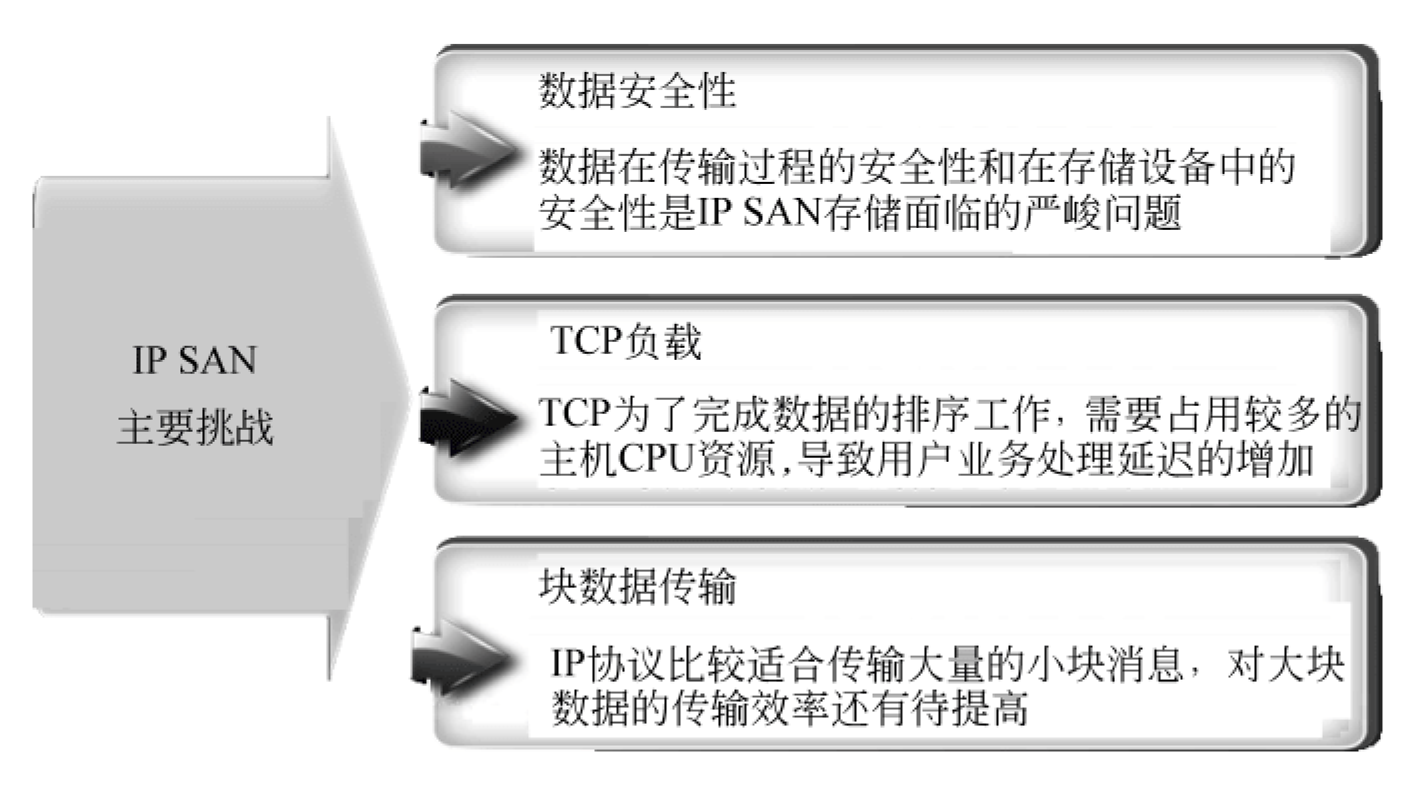


图 3-33　IP SAN 面临的挑战

5．FC SAN 与 IP SAN 的比较

FC SAN 与 IP SAN 的比较如表 3-3 所示。

表 3-3　FC SAN 与 IP SAN 的比较

描　述	FC SAN	IP SAN
网络速度	1Gbps、2Gbps、4Gbps、8Gbps	1Gbps、10Gbps
网络架构	单独建设光纤网络和 HBA 卡	使用现有 IP 网络
传输距离	受到光纤传输距离的限制	理论上没有距离限制
管理、维护	技术和管理较复杂	与 IP 设备一样操作简单
兼容性	兼容性差	与所有 IP 网络设备都兼容
性能	非常高的传输和读写性能	目前主流传输速率为 1Gbps，并正在发展
成本	购买(光纤交换机、HBA 卡、光纤磁盘阵列等)、维护(培训人员、系统设置与监测等)成本高	与 FC-SAN 相比，购买与维护成本都较低，有更高的投资收益比例
容灾	容灾的硬件、软件成本高	本身可以实现本地和异地容灾，且成本低
安全性	较高	较低

3.3.6　IP SAN 的组成和组网连接

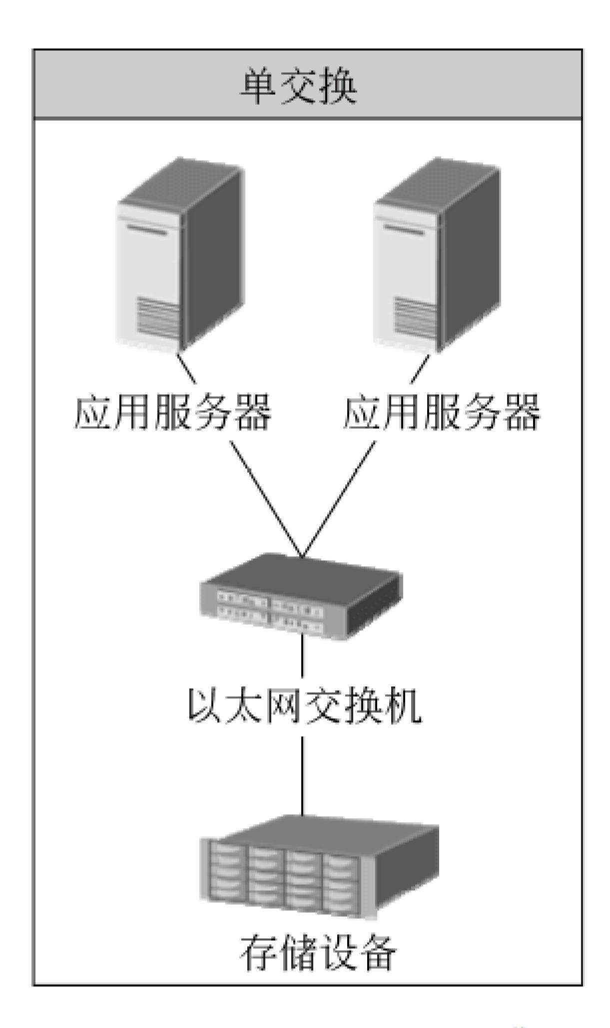
1. IP-SAN 典型组网

(1) 直连：主机与存储之间直接通过以太网卡、TOE 卡或 iSCSI HBA 卡连接，这种组网方式简单、经济，但较多的主机分享存储资源比较困难。

(2) 单交换：主机与存储之间由一台以太网交换机，同时主机安装以太网卡或 TOE 或 iSCSI HBA 卡以实现连接。这种组网结构使多台主机能共同分享同一台存储设备，扩展性强，但交换机处存在单点故障。

(3) 双交换：同一台主机到存储阵列端可由多条路径连接，扩展性强，避免了在以太网交换机处形成单点故障，如图 3-34 所示。





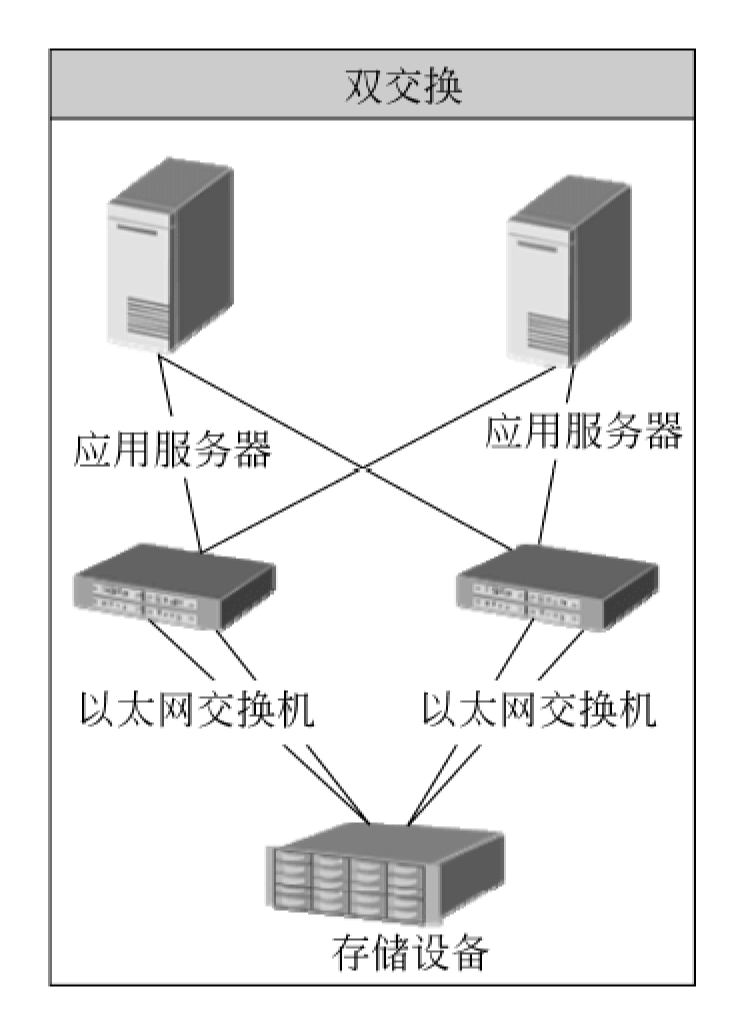


图 3-34　IP-SAN 典型组网

2. IP-SAN 的组件

IP-SAN 基于十分成熟的以太网技术，由于设置配置技术简单、低成本的特色相当明显，而且普通服务器或 PC 只需要具备网卡即可共享和使用大容量的存储空间。由于是基于 IP 协议的，所以能容纳所有 IP 协议网络中的部件。用户可以在任何需要的地方创建实际的 SAN 网络，而不需要专门的光纤通道网络在服务器和存储设备之间传送数据。同时，因为没有光纤通道对传输距离的限制，IP-SAN 使用标准的 TCP/IP 协议，数据即可在以太网上进行传输。IP-SAN 网络对于那些要求流量不太高的应用场合以及预算不充足的用户是一个非常好的选择，如图 3-35 所示。

IP-SAN 由① iSCSI 存储设备；②以太网交换机；③以太网卡和 iscsi initiator 软件；④以太网网线组成。

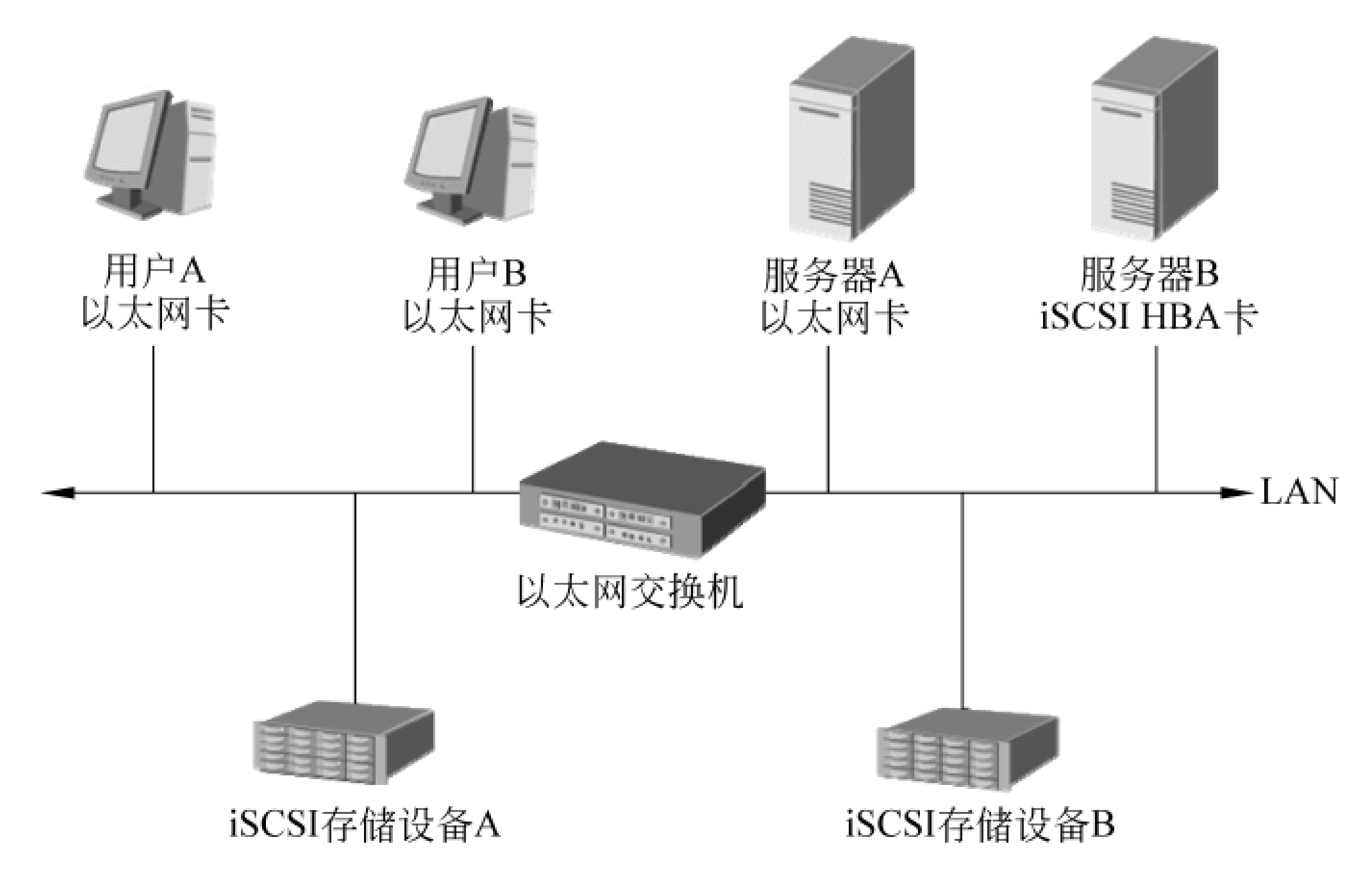


图 3-35 IP-SAN 的组件

3. iSCSI 连接方式

iSCSI 设备通常使用 IP 接口作为其主机接口，并可以通过与传统以太网交换机的连接，构建一个基于 TCP/IP 协议的存储区域网络。根据主机端所采用的连接方式的不同，iSCSI 设备与主机的连接通常有三种形式。

(1) 以太网卡＋Initiator 软件方式：采用这种方式的主机使用标准的以太网卡(NIC)与网络进行连接。iSCSI 层和 TCP/IP 协议栈功能通过主机 CPU 运行软件计算完成。由于这种方式直接使用传统主机系统通用的 NIC 卡，所以成本最低，但是由于需要占用 CPU 资源进行 iSCSI 协议和 TCP/IP 协议处理，所以会导致主机系统性能的下降。

(2) TOE＋Initiator 软件方式：采用这种方式的主机使用 TOE(TCP offload Engine，TCP 卸载引擎)网卡，iSCSI 协议的功能仍然由主机的 CPU 完成，但是 TCP 协议处理则交由 TOE 网卡完成，从而有效减轻了主机端的负担。

(3) iSCSI HBA 卡方式：采用这种方式的主机，其 iSCSI 协议功能及 TCP/IP 协议栈功能均由 iSCSI HBA 卡完成，对主机的开销占用最小。

IP-SAN 根据主机与存储的连接方式不同，可以分为三种，如图 3-36 所示。

4. 以太网卡＋Initiator 软件实现方式

服务器、工作站等主机设备使用标准的以太网卡，通过以太网线直接与以太网交换机连

接，iSCSI 存储也通过以太网线连接到以太网交换机上，或直接连接到主机的以太网卡上。在主机上安装 Initiator 软件以便将以太网卡虚拟为 iSCSI 卡，用以接收和发送 iSCSI 数据报文，从而实现主机和 iSCSI 设备之间的 iSCSI 协议和 TCP/IP 协议传输功能。由于采用普通的标准以太网卡和以太网交换机，无须额外配置适配器，因此此种方式硬件成本最低。缺点是进行 iSCSI 包文和 TCP/IP 包文转换需要占用主机端的资源，使主机的运行开销增加而导致系统性能下降。不过在对于 I/O 和带宽性能要求较低的应用环境中基本能够满足数据访问要求，如图 3-37 所示。



图 3-36　IP-SAN 存储连接方式

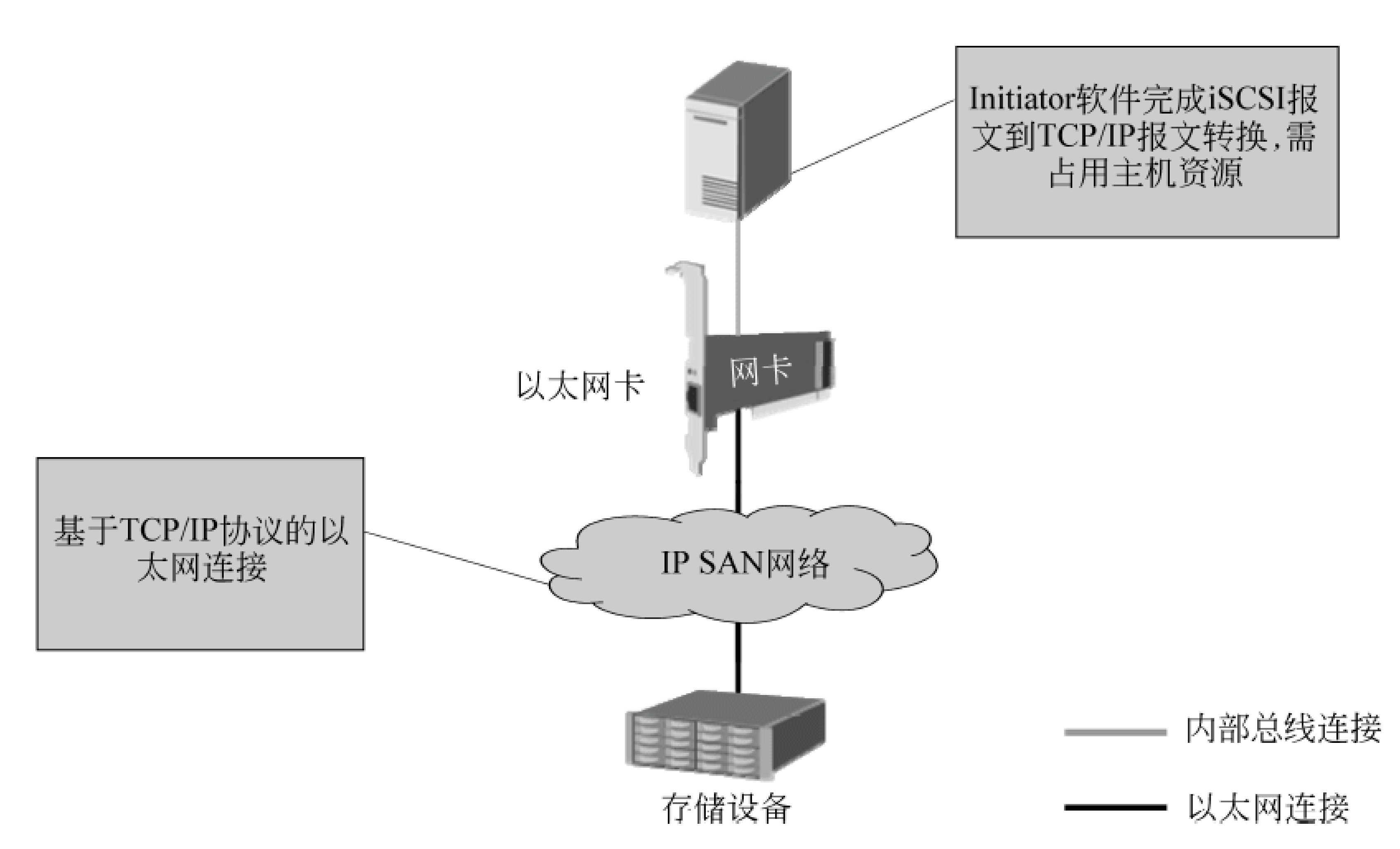


图 3-37　以太网卡＋Initiator 软件实现方式

5. TOE 网卡＋Initiator 软件实现方式

智能以太网卡可以将网络数据流量的处理工作全部转到网卡上的集成硬件中进行，TCP/IP 协议栈功能由 TOE 网卡完成，而 iSCSI 层的功能仍旧由主机来完成，由此，采用 TOE 网卡可以大幅度提高数据的传输速率。与纯软件的方式相比较而言，这种方式部分降低了主机系统的运行开销而又不会使网络构建成本过多增加，是一种比较折中的配置方案，如图 3-38 所示。

6. iSCSI HBA 卡连接方式

在主机上安装 iSCSI HBA 适配卡，从而实现主机与交换机之间、主机与存储设备之间的高效数据交换。iSCSI 层和 TCP/IP 协议栈的功能均由主机总线适配器（HBA）来完成，对主机 CPU 的占用最少，这种方式数据传输性能最好，但是系统构建价格也最高，如图 3-39 所示。

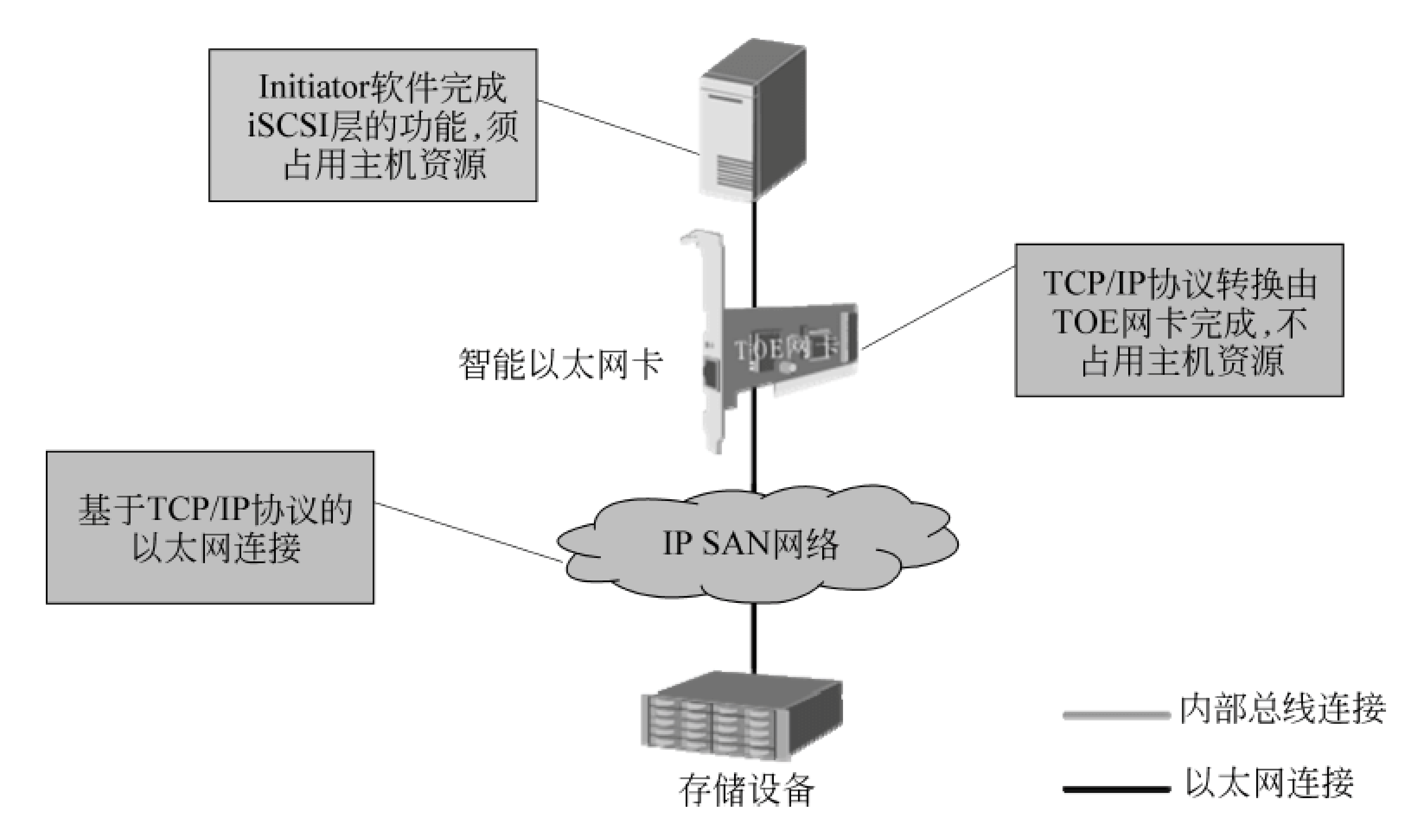


图 3-38　TOE 网卡＋Initiator 软件实现方式

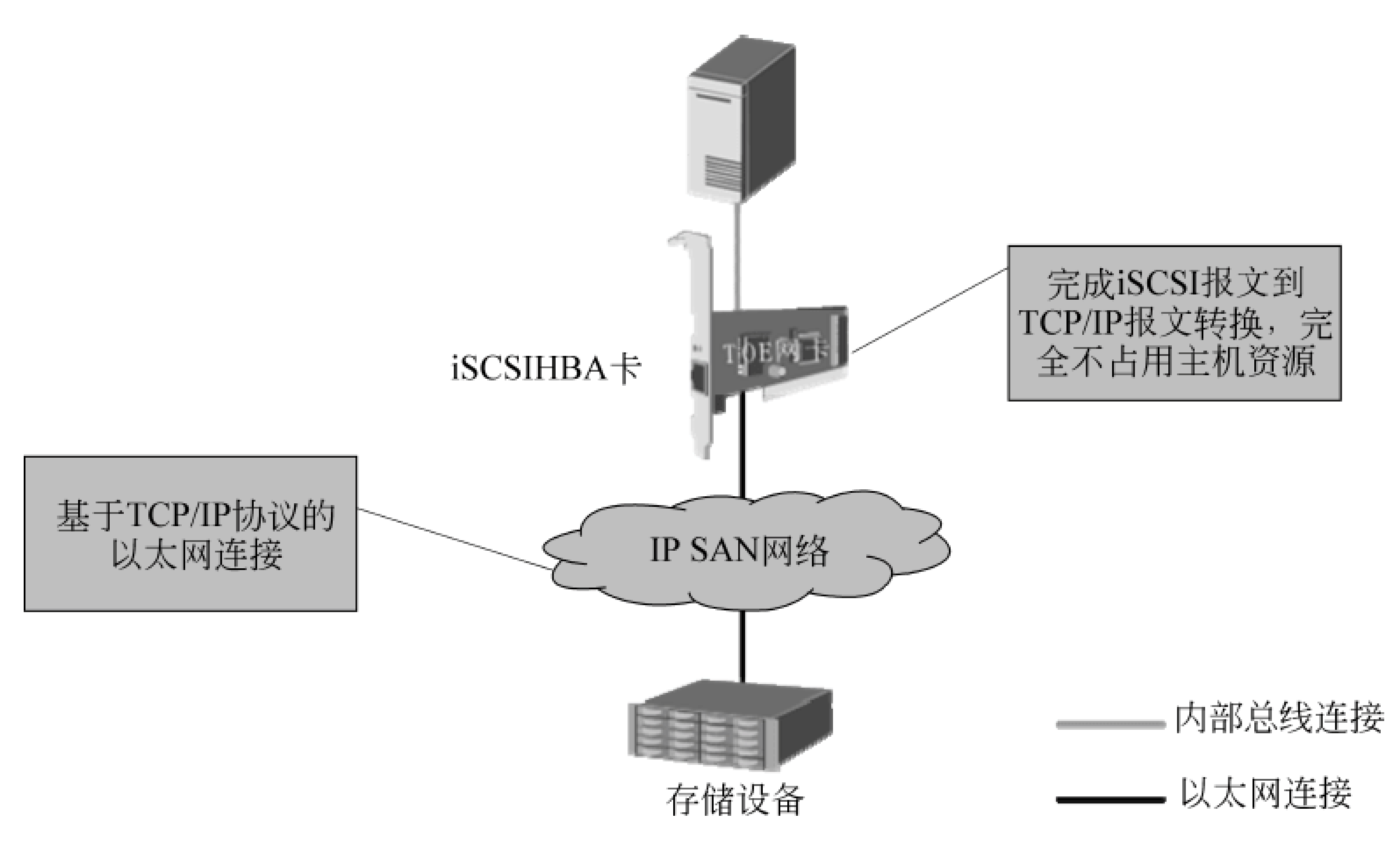


图 3-39　TOEiSCSI HBA 卡连接方式

3.3.7　FC 协议与 TCP 协议融合

1. FC-SAN 与 IP-SAN 在产品上的融合

如磁盘阵列上控制器上既包含 FC 端口，又包含 iSCSI 口，能同时满足 FC-SAN 的组网需求，也能满足 IP-SAN 的组网需求，这是不完全的融合，如图 3-40 所示。

2. FC-SAN 与 IP-SAN 解决方案的融合

统一存储解决方案典型 T 系列产品，可以同时对外提供 FC-SAN、IP-SAN 和 NAS 共享，如图 3-41 所示。

3. FC 与 TCP 协议融合

基于 IP 的光纤通道(Fibre Channel over IP，FCIP)是连接 TCP/IP 网络链路上的光纤通道架构的一项 IETF 标准。利用目前的 IP 协议和设施来连接两个异地 FC SAN 的隧道，用以解决两个 FC SAN 的互联问题。

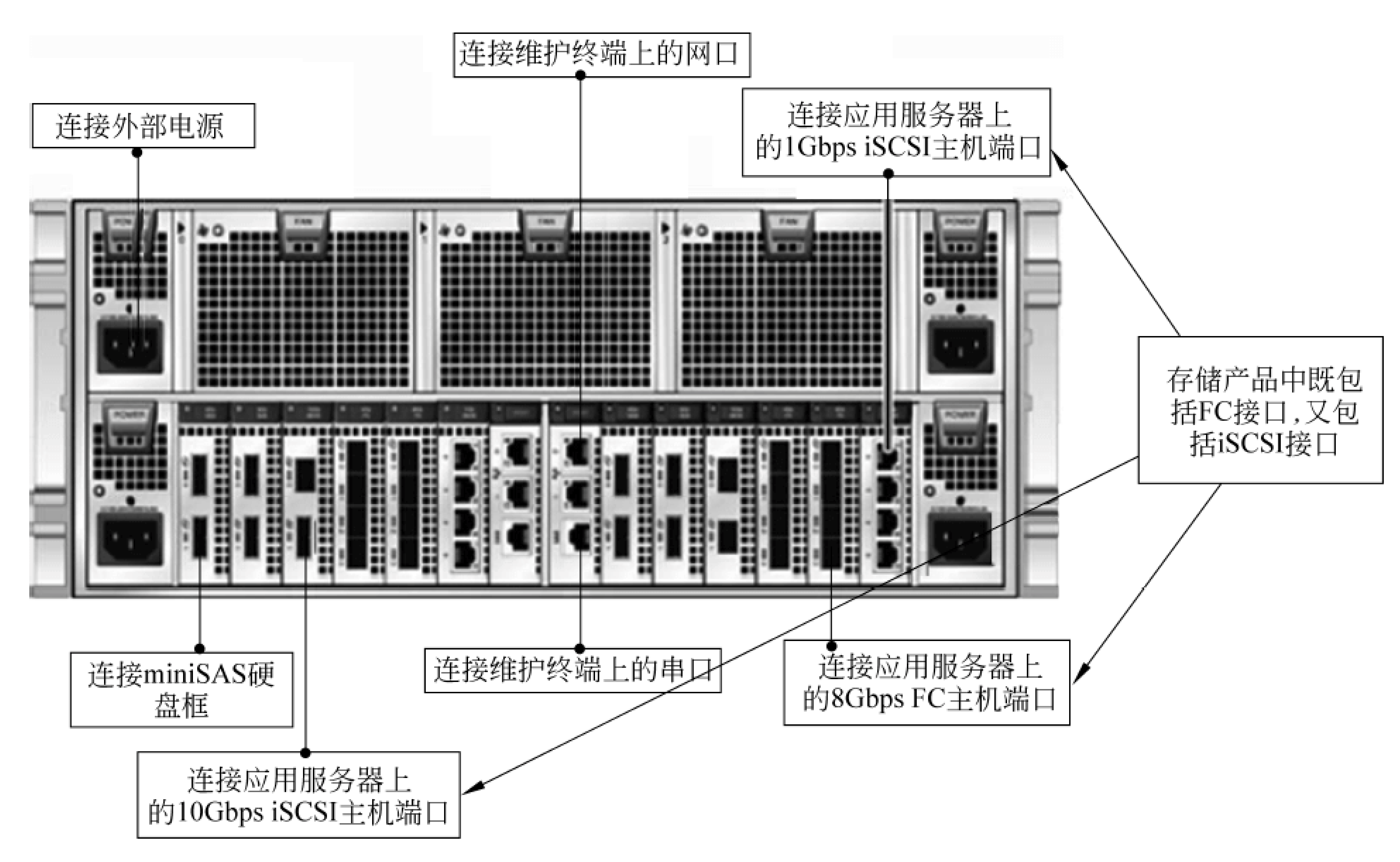


图 3-40　磁盘阵列控制器

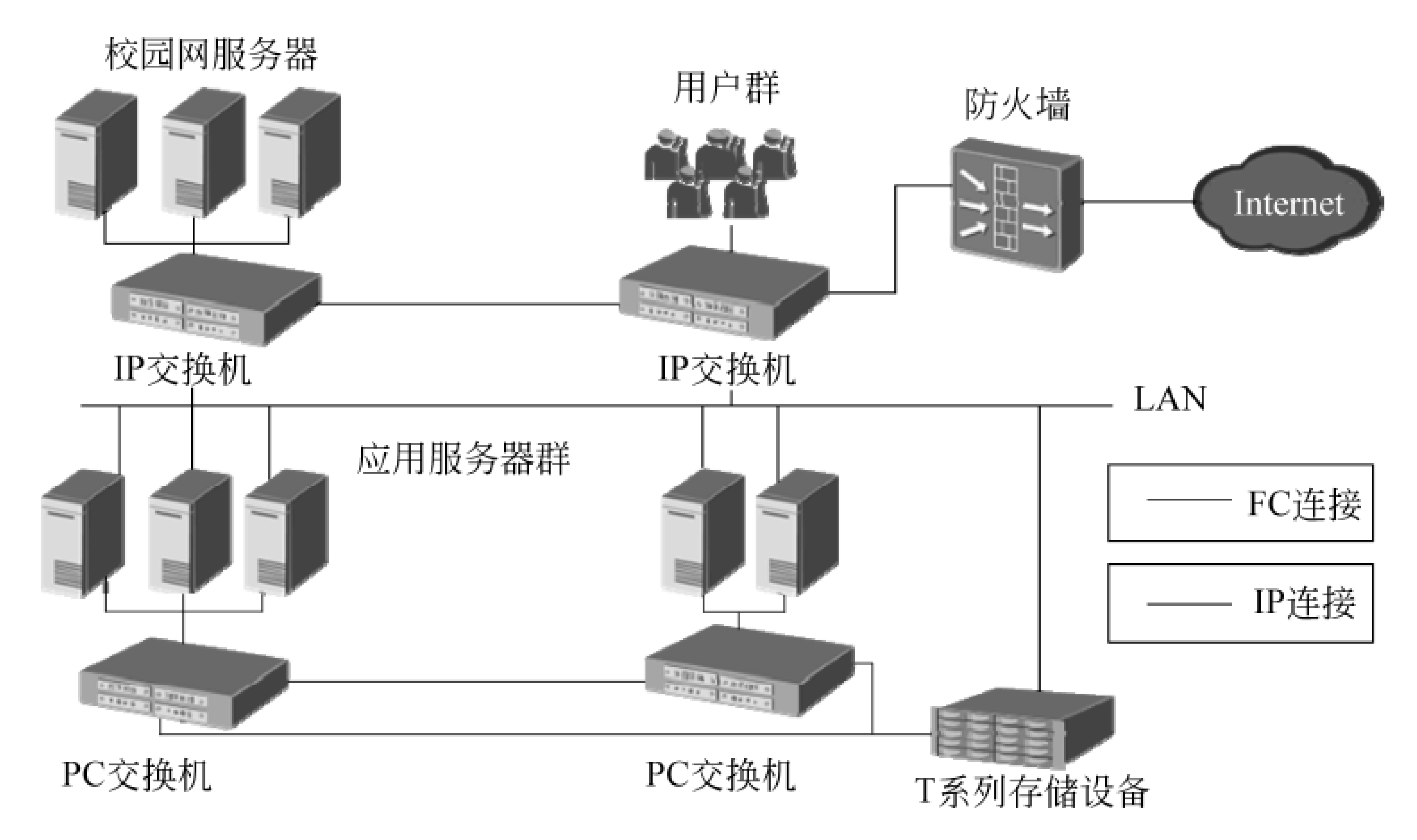


图 3-41　FC-SAN 与 IP-SAN 解决方案

Internet 光纤信道协议(Internet Fibre Channel Protocol,iFCP)是一种网关到网关的协议,为 TCP/IP 网络上的光纤设备提供光纤信道通信服务。iFCP 使用 TCP 提供拥塞控制、差错监测与恢复功能。iFCP 主要目标是使现有的光纤信道设备能够在 IP 网络上以线速互联与组网。此协议及其定义的帧地址转换方法允许通过透明网关(Transparent Gateway)将光纤信道存储设备附加到基于 IP 的网络结构。

FCoE(Fibre Channel over Ethernet)利用以太网路传送光纤通道(Fibre Channel)的信号,让光纤通信的资料可以在 10Gbps 骨干以太网络中传输,但仍然是使用光纤通道的协定。

基于 IP 的光纤通道(IP over Fiber Channel,IPFC)使用在两个服务器之间的光纤通道

连接作为 IP 数据交换的媒介。为此，IPFC 定义了如何通过光纤通道网络传送 IP 分组。对本地 IP 配置的连接使用 ifconfig 或 ipconfig，然后 IPFC 驱动器寻址光纤通道主机适配卡，就可以在光纤通道上发送 IP 分组。

下面我们着重介绍一下 FCIP 和 iFCP 协议。

目前 FC 与 TCP/IP 协议的真正融合主要有两种趋势。

(1) TCP/IP 网络承载 FC 信道：FCIP、iFCP、FCOE。

(2) 以 FC 信道承载 TCP/IP 数据：IPFC。

从现有的情况来看，以太网技术和 FC 技术都在飞速发展，IP-SAN 和 FC-SAN 会在很长的一段时间内都将是并存且互为补充的。

4. FCIP 协议

基于 IP 协议的光纤通道(Fiber Channel over IP，FCIP)是基于 IP 协议传输的光纤通道数据帧的解决方案，是由 Brocade、Gadzoox、Lucent、McData 以及 Qlogic 公司共同提出的。FCIP 这一技术的核心，是把光纤通道协议的数据帧封装在 IP 数据包里，以便在覆盖广阔的 TCP/IP 网络中进行传递。网络中的其他设备接收后，由专门的目标设备进行解封装，以便还原 FC 数据帧。FCIP 协议实质上就是采用隧道技术的 IP-SAN 方案。采用 FCIP 技术可以实现利用目前的 IP 网络来联接两个异地的 FC-SAN，以解决 FC-SAN 之间的互联问题。这一隧道传输技术是通过使用 FCIP 网关来实现的，通过光纤通道交换机的扩展端口连接到每个 FC-SAN 上，所有发往远程站点的存储数据均通过公用的 IP 隧道。接收端的光纤通道交换机负责将到来的每个帧交换至目的光纤通道端点设备。

FCIP 协议是一种隧道(tunnel)协议，IP 地址和 TCP 连接只用在位于 IP 网络重点的 FCIP 网关设备上。FCIP 能够为两个 FC-SAN 之间提供 IP 连接，但是不能为两个独立的 FC 存储设备提供 IP 连接，即 FCIP 不能通过 IP 协议实现 FC 存储设备端到端的连接。

FCIP 协议利用 IP 网络中创建的“隧道”在两个 FC-SAN 网络之间实现 FC 协议的数据传输，将真正意义上的远程数据镜像和 FC-SAN 的灵活性以及 IP 网络的低成本和易用性结合在一起，降低了远程操作的成本和操作的复杂性。FCIP 提供了在 TCP/IP 协议中封装 FC 协议数据帧的方法，消除了 FC 目前存在的距离限制，允许通过 IP 网络来互联FC-SAN，使得数据的访问变得更加灵活，存储策略的部署更加容易，如图 3-42 所示。

5. FCIP 的协议栈

FCIP 协议是一个点到点的隧道封装协议，它可以实现多个本地 FC-SAN 网络经由 FCIP 网关通过 IP 网络进行互联并对其进行管理。在 FCIP 的协议栈中，FCIP 协议处于 FC 和 TCP 之间，也就意味着 FCIP 可以互联 FC 和 TCP 这两种协议网络。在 TCP 下层是 IP 协议和下层的数据链路以及物理层协议，而 FC 协议的上层则有 FCP 和 SCSI 协议，由此可见 FCIP 协议联系了底层的 IP 网络和高层的 SCSI 应用，实现了不同网络、不同协议之间的网络设备互联和应用的融合。

在 FCIP 的协议栈中，FCIP 协议处于 FC 和 TCP 之间，也就意味着 FCIP 可以互联 FC 和 TCP 这两种协议网络，如图 3-43 所示。

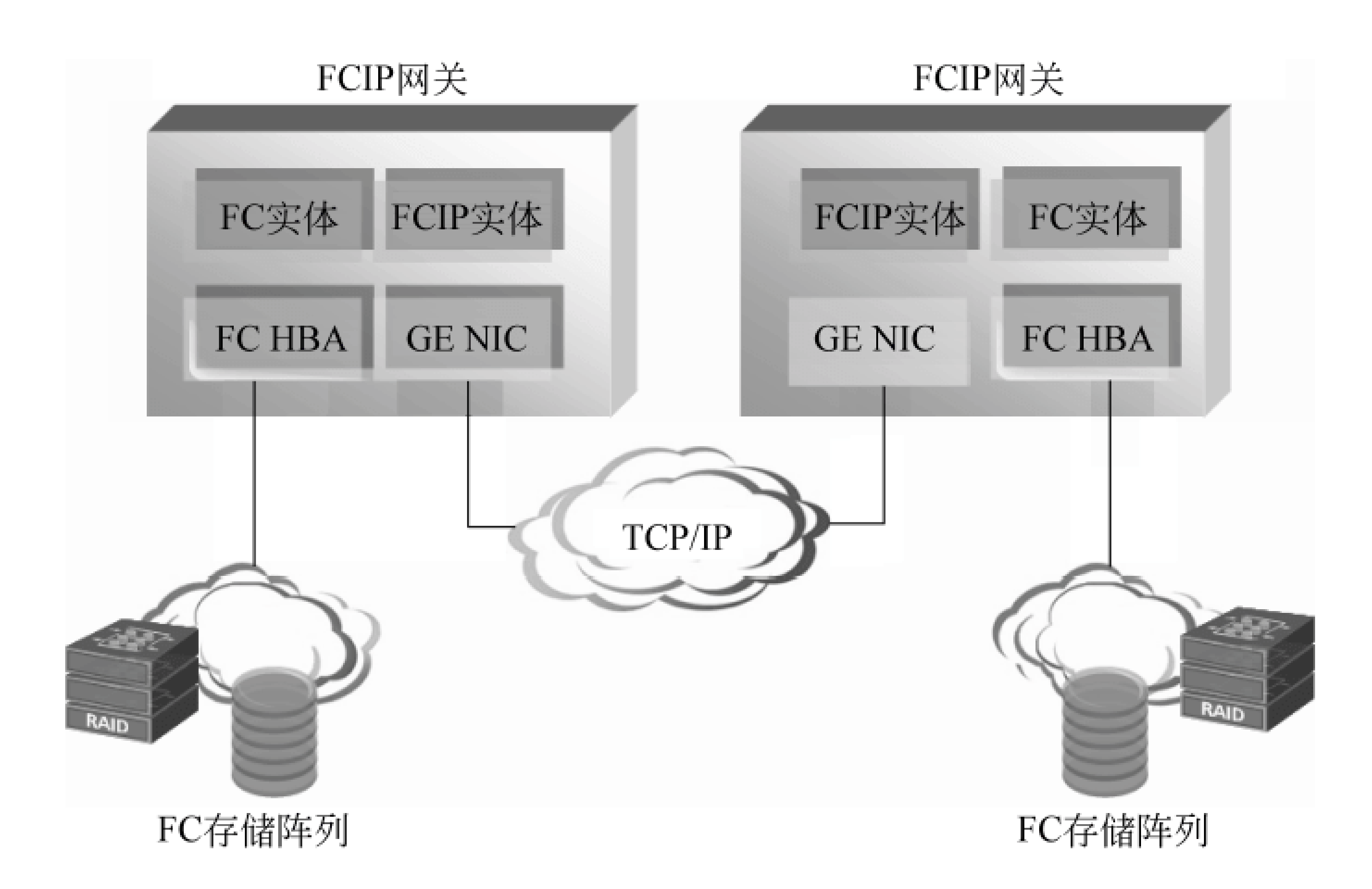


图 3-42　基于 IP 协议传输的光纤通道数据帧的解决方案

SCSI应用（文件系统、数据库）		
SCSI块指令	SCSI流指令	其他SCSI指令
SCSI指令、数据和状态		
FCP		
FC		
FCIP		
TCP		
IP		
以太网		

图 3-43　FCIP 的协议栈

6. FCIP 的数据封装

在 FCIP 数据封装中，光纤通道网络体系结构提供的终端寻址、地址解析、信息路由等均保持不变，而 IP 协议在这里只是作为传输协议用以承载 FC 数据帧在 IP 网络中进行传输。

FCIP 在 FC 帧和 TCP 包头之间加入了 FCIP 包头，用来显示 FCIP 协议的版本、帧长度等字段。发送端 FCIP 网关设备将 FC 封装为 FCIP 帧，通过 IP 网络传送。接收端 FCIP 网关设备接收到 FCIP 帧后，解封装 IP 和 TCP 包头，还原成 FC 帧并通过一个或多个 FC 交换机发送到目的节点。

FCIP 协议是一个隧道协议，它提供把 FC 协议数据帧封装进 IP 包以便在 IP 网络中进行传输的方法，如图 3-44 所示。

7. FCIP 通信原理

整个 FCIP 的通信过程是由其数据引擎推动进行的。首先在源 FCIP 连接端点（FCIP Link EndPoint，FCIP LEP）处对 FC 协议数据帧进行封装，然后通过 TCP/IP 协议在 IP 网

络中进行传输，到达目的 FCIP 连接端点后进行解封装，读出其中的数据并执行其中的 FC 指令。

FCIP 作为一种隧道技术，仍然存在一些缺点。首先，其带宽相对 FC 而言，由于利用的是 IP 通道，所以带宽仍远低于 FC。其次，由于 FC 协议帧被封装进 IP 数据包中，但是 IP 网络智能管理工具并不能识别这些数据，使得很多很好的管理控制机制无法应用于 FCIP，比如目录服务、流量控制和 QoS 等。最后，由于 FCIP 仅仅是在 TCP/IP 网络中构建起一个传输 FC 协议帧的隧道将两个远端的 FC-SAN 连接起来，它并没有解决单个 FC-SAN 的设备互操作性问题和管理问题，本地的 SAN 采用的仍然是 FC 技术，如图 3-45 所示。

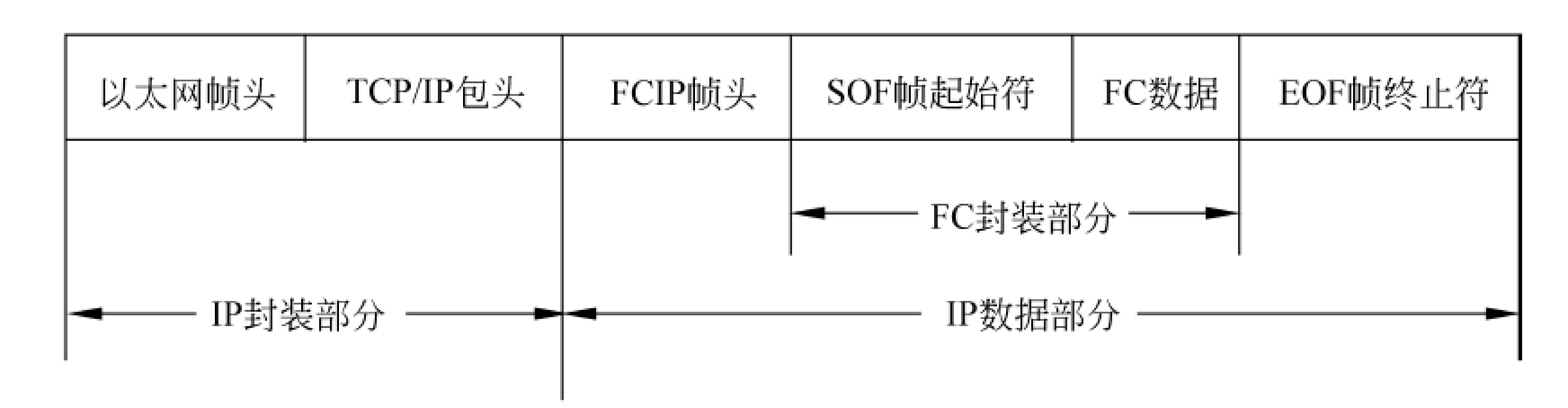


图 3-44　FCIP 的数据封装

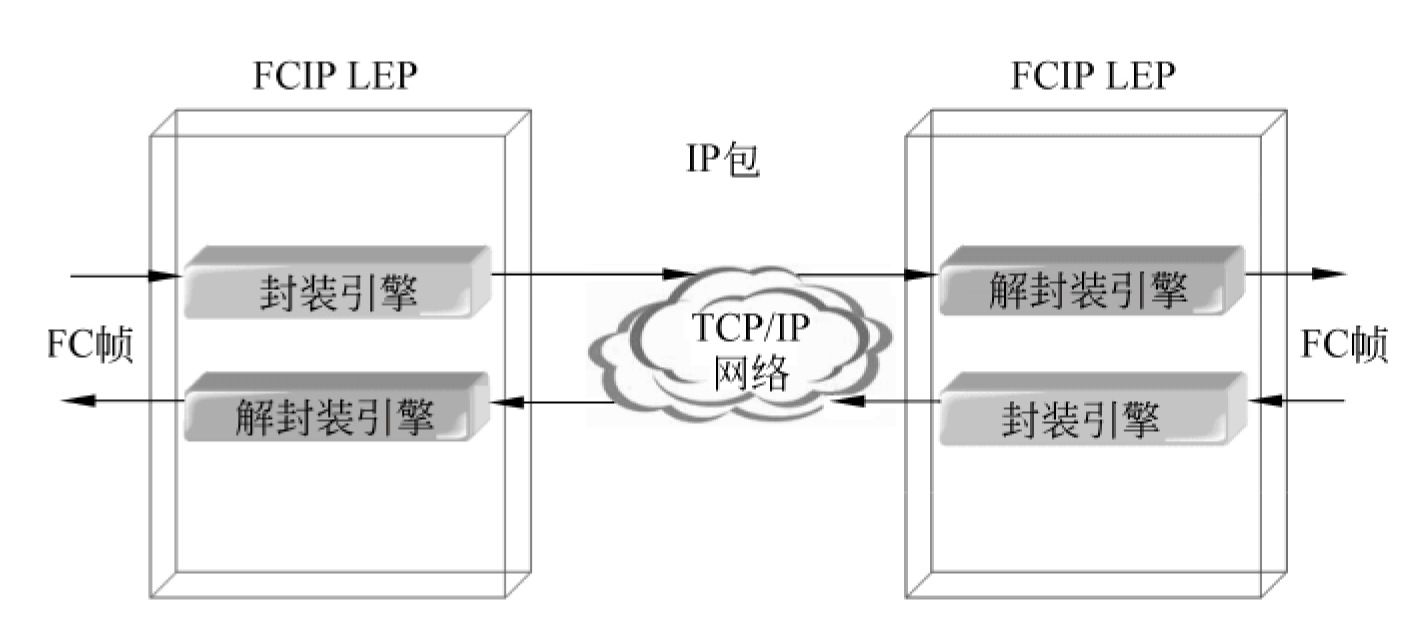


图 3-45　4 FCIP 通信原理

8. iFCP 协议

iFCP 如图 3-46 所示。iFCP 可以直接替代 FC 架构，通过 iFCP 存储交换机可以直接连接 FC 的各个设备并进行分组，而不仅仅是简单地对 FC-SAN 进行远程连接，但是 iFCP 不支持独立的存储区域网络的合并操作(Merge)，因此无法组成单一的逻辑 SAN。iFCP 的优势在于在建立连接的同时还能够建立网关分区，可以将出现故障的区域隔离开来，并克服了点到点隧道的限制。并且 iFCP 提供 FC 设备端到端的连接，TCP 连接的中断只会影响到一个通信对，而不会影响到其他通信，也不会将一个设备的错误带给其他设备。基于 iFCP 实现了 SAN 的路由故障隔离、安全及灵活管理，具有比 FCIP 更高的可靠性。

9. iFCP 的协议栈

iFCP 协议层的主要功能是在本地和远程 N_PORT 间传输光纤通道帧映像。当帧被传输到远程 N_PORT 时，iFCP 层开始封装并路由光纤通道帧。光纤通道帧包括每一个光纤通道信息单元，通过预先建立的 TCP 连接在 IP 网络上传输。

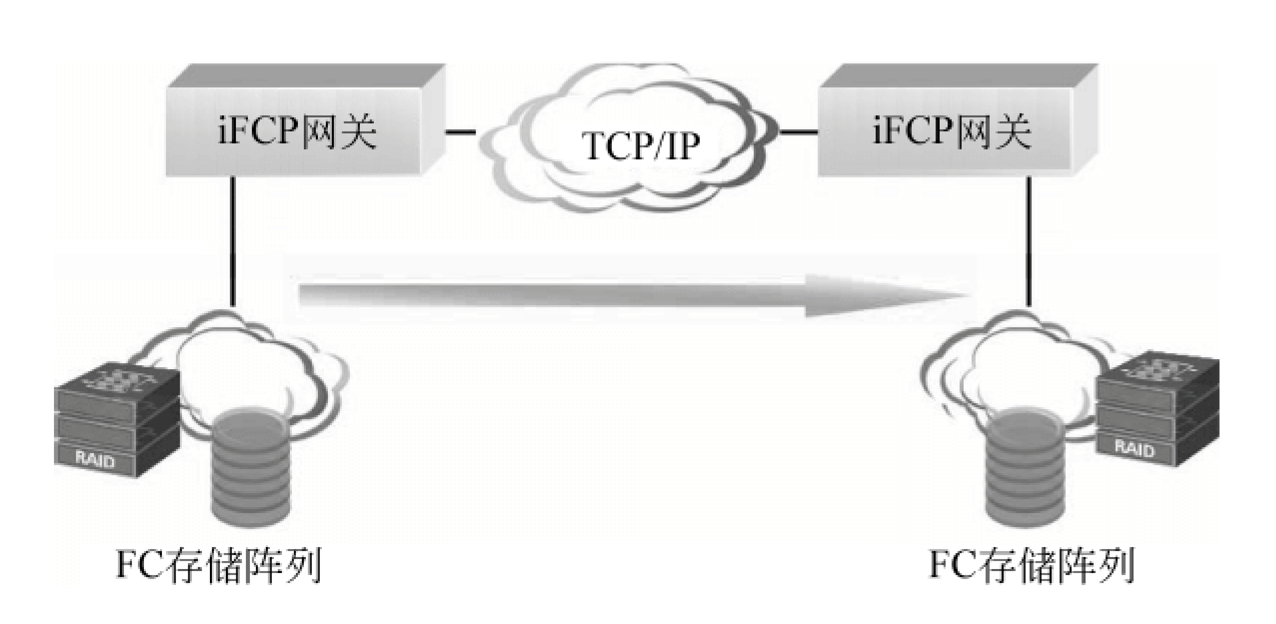


图 3-46 iFCP 协议

iFCP 协议位于 TCP/IP 协议和 FC 协议之间，可以起到连接这两种协议网络的作用，如图 3-47 所示。

SCSI应用（文件系统、数据库）		
SCSI块指令	SCSI流指令	其他SCSI指令
SCSI指令、数据和状态		
FC		
iFCP		
TCP		
IP		
以太网		

图 3-47 iFCP 协议栈

10. iFCP 协议封装

iFCP 在 FC 帧和 TCP 包头之间，在 iFCP 层中，FC 设备的 24 位 fabric 地址被映射到一个唯一的 IP 地址上，为 Fibre Channel 启动器和目标提供了本地 IP 地址的编址工作。iFCP 代替了 Fibre Channel 的底层传输层（FC-2），它使用 TCP/IP 在 IP 网络上进行可靠传输，如图 3-48 所示。

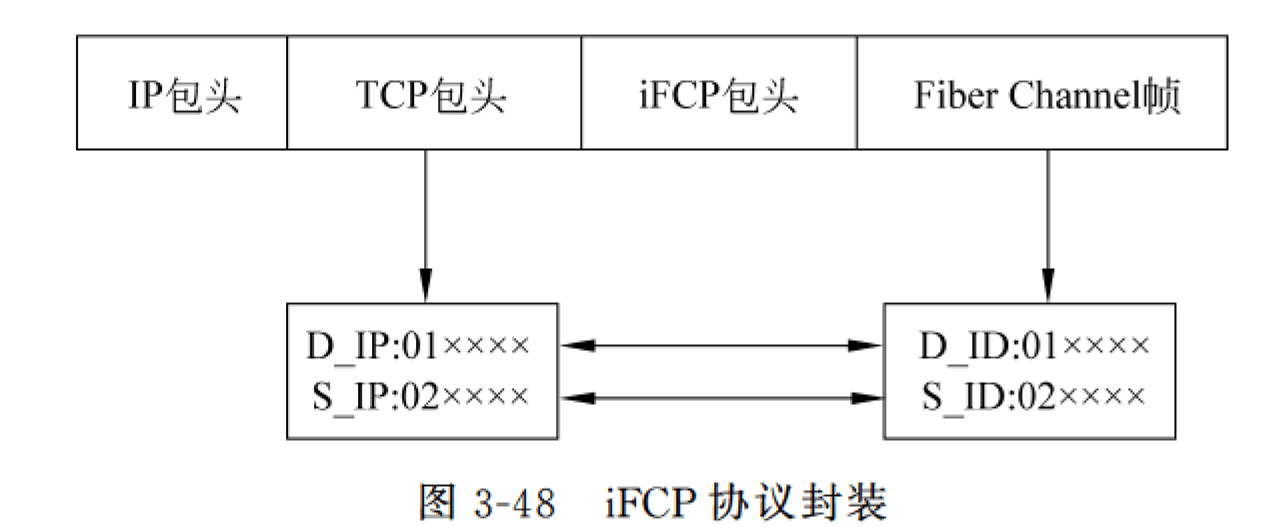


图 3-48 iFCP 协议封装

11. iFCP 的工作原理

iFCP 的工作原理是，将 Fibre Channel 数据以 IP 包形式封装，并将 IP 地址映射到分离光纤通道设备。由于在 IP 网中每类光纤通道设备都有其独特标识，因而能够与位于

IP 网其他节点的设备单独进行存储数据收发。光纤通道信号在 iFCP 网关处终止，信号转换后存储通信在 IP 网中进行，这样 iFCP 就打破了传统光纤通道网的距离（在不使用中继器的情况下，FC 的传输距离约为 10km）限制，如图 3-49 所示。

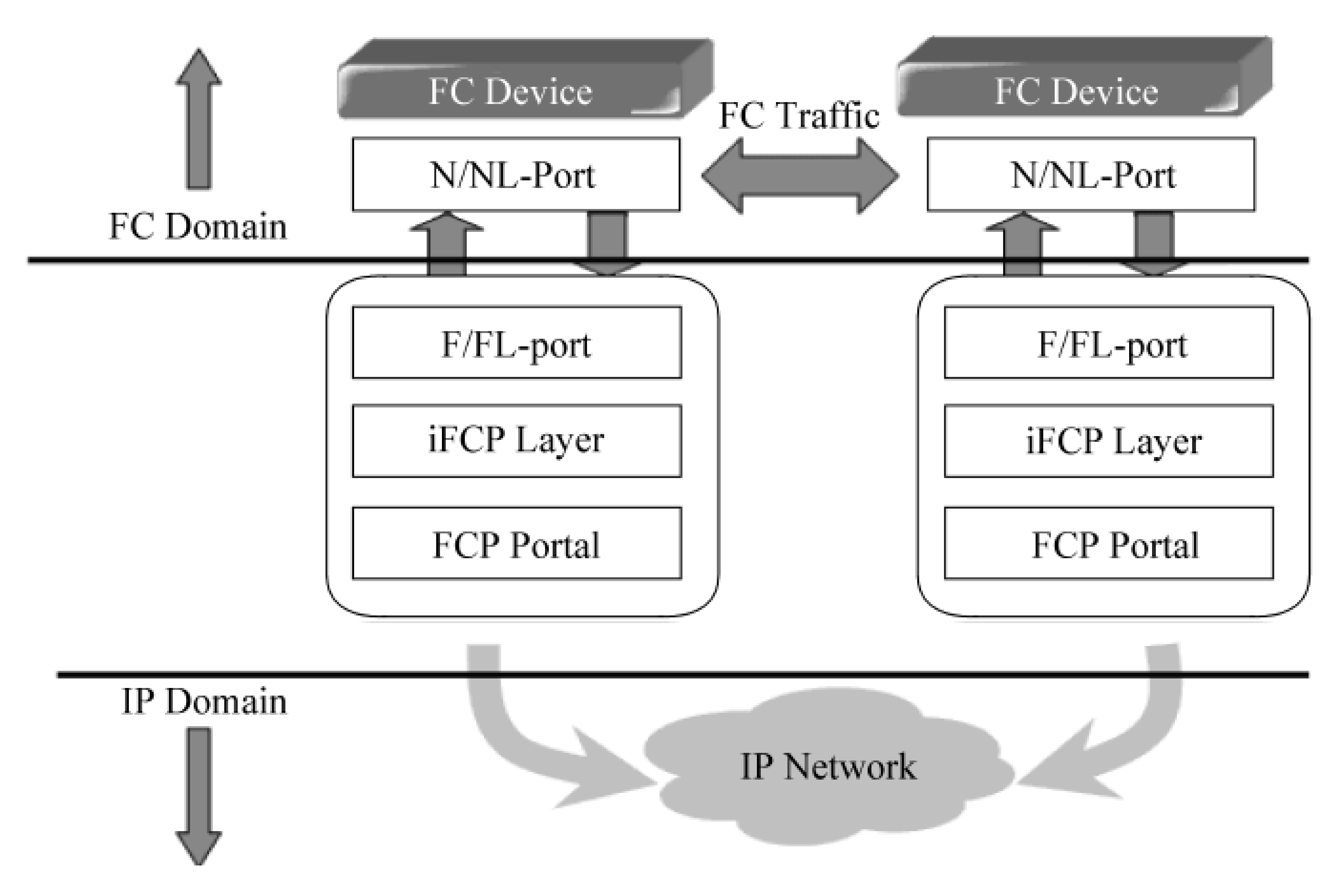


图 3-49　iFCP 工作原理

12. iFCP 网络体系架构

在使用 iFCP 协议构建的 IP-SAN 存储网络中，存储设备没有被限制在光纤通道 SAN 的 IP 网络中分布。iFCP 存储交换机直接接替 FC-SAN 中的光纤通道交换机，这就意味着 iFCP 交换机也具有 SNS（存储名称服务器）功能，为终端节点提供名称发现服务。在 iFCP 交换机中指派 4 字节的 IP 地址给每一个光纤通道终端节点。当光纤通道设备发送一个 SNS 名称查询时，这个请求将被 iFCP 交换机截住，并由 iSNS 服务器进行解释。在光纤通道层，一个适用的目标地址表将返回给发起者，此时其余 IP 的光纤通道地址表就映射光纤通道地址，以便相应的 IP 地址可以通过 IP 网络传到目标设备，如图 3-50 所示。

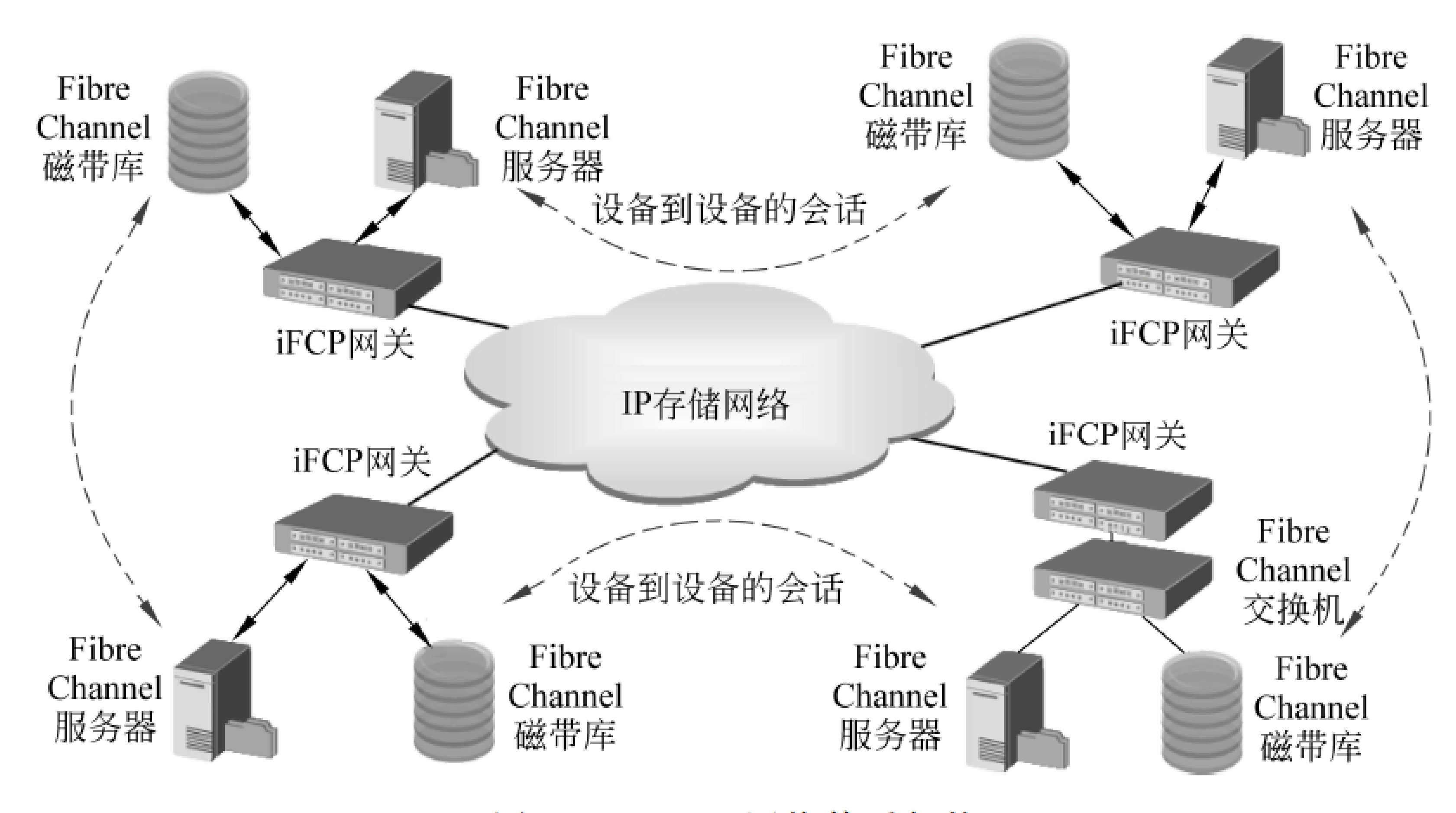


图 3-50　iFCP 网络体系架构

13. iSCSI 协议、FCIP 协议、iFCP 协议比较

FCIP 和 iSCSI 技术在 IP 存储领域是两个相对的极端，FCIP 可以看作是 FC 的扩展策略，它包含了部分的 IP 协议信息和大量的 FC 成分，所以从严格意义上来说 FCIP 并不能算真正的 IP 存储策略。而 iSCSI 协议的目的则是要用 IP 协议完全取代 FC 协议在 SAN 中的应用，iSCSI 协议中完全不含有 FC 的内容，只包含了 IP 信息。iSCSI 与 iFCP 相比较具有一定的相似性，iSCSI 和 iFCP 都在存储端设备中采用了 IP 协议技术，而不同在于，iSCSI 为通过 IP 网络进行块数据传输定义了其本身的串行 SCSI 的实现。这三种协议都位于 TCP/IP 和 SCSI 协议之间，为 TCP/IP 和 SCSI 的沟通起到了纽带的作用，如图 3-51 所示。

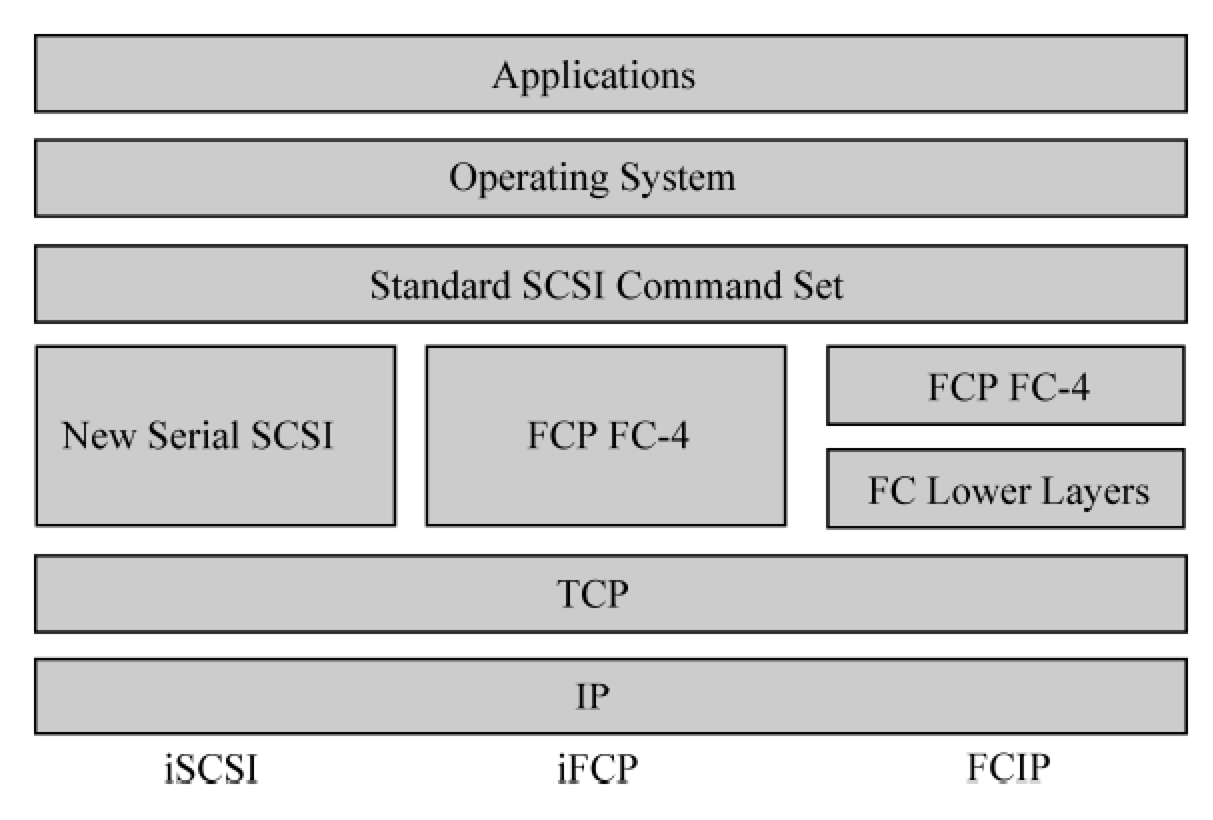


图 3-51　iSCSI 协议、FCIP 协议、iFCP 协议的比较

14. FCoE 协议

FCoE(Fibre Channel over Ethernet)可以提供标准的光纤通道原有服务，如发现、全局名称命名、分区等，而且这些服务都可以照标准原有的运作，保有 FC 原有的低延迟性、高性能。

从 FC 协议的角度来看，FCoE 就是把 FC 承载在一种新型的链路上，即以太网二层链路；从以太网的角度来看，FCoE 仅是其承载的另外一种上层协议(类似于 IP/IPX)。直接在增强型无损以太网基础设施上传输光纤信道信号功能的协议。FCoE 把 FC 帧封装在以太网帧中，允许 LAN 和 SAN 的业务流量在同一个以太网中传送，如图 3-52 所示。

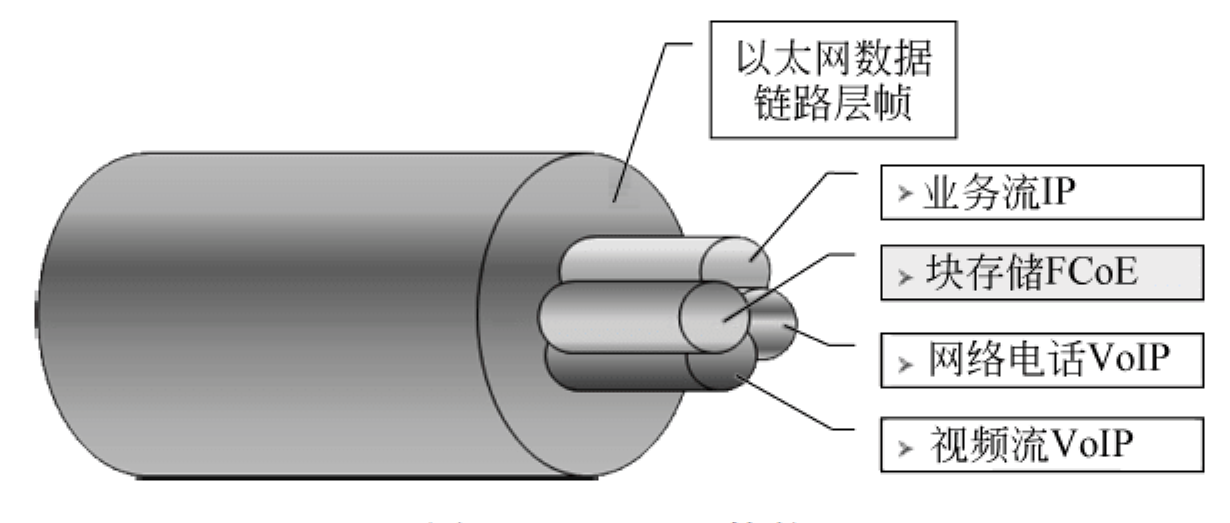


图 3-52　FCoE 协议

15. FCoE 协议的封装

FC 网络协议栈有五层，其中 FC0 定义承载介质类型，FC1 定义帧编解码方式，FC2 定

义分帧协议和流控机制，FC3 定义通用服务，FC4 定义上层协议到 FC 的映射。

FCoE 是把 FC-2 层以上的内容封装到以太网报文中进行承载，如图 3-53 所示。

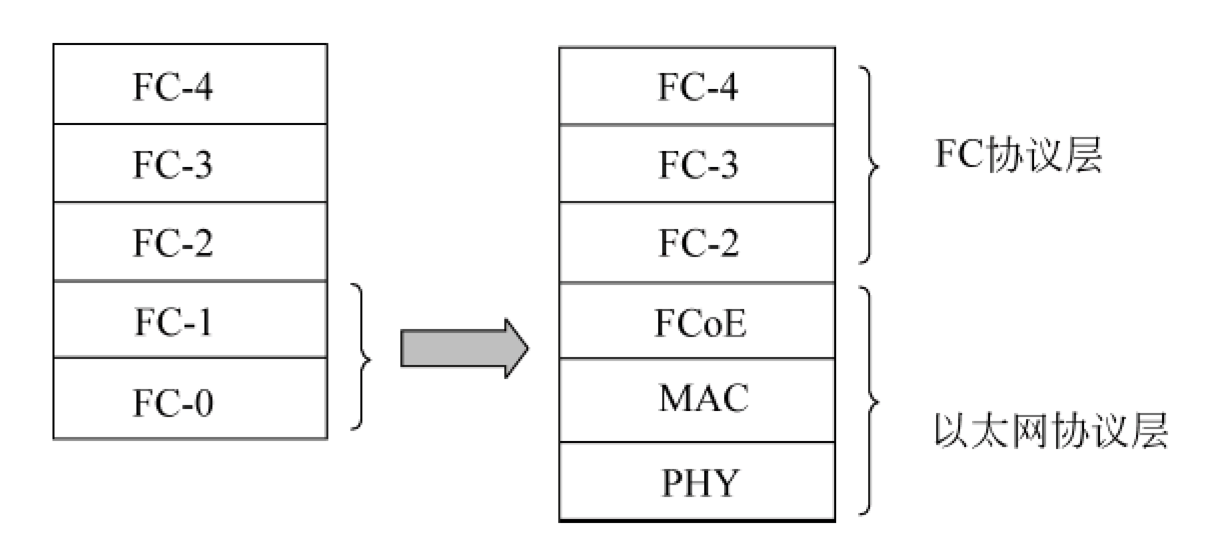


图 3-53　FCoE 协议的封装

16. 融合增强型以太网（CEE）

传统的以太网是一种尽力服务的网络模式，当网络拥塞时有可能发生丢包，进而导致出现数据包重传或超时现象。FCoE 网络融合技术的出现，对以太网提出了无丢包服务的要求。为此，IEEE 802.1 和 IETF 标准组织制定一些新的标准，创建一个新的、更强大的以太网协议系列，即融合增强型以太网（CEE）。

（1）基于优先级的流量控制（PFC）

以太网 Pause 机制能够实现网络不丢包的要求，但它会阻止一条链路上的所有流量，PFC 是对 Pause 机制的一种增强。PFC 可以在一条以太物理链路上创建 8 个独立的虚拟链路，并允许单独暂停和重启其中任意一条虚拟链路。这一方法使网络能够为单个虚拟链路创建无丢包类别的服务，使其能够与同一接口上的其他类型的流量共存。

（2）增强的传输选择（ETS）

ETS 可以为不同的业务流量设定优先级和保证带宽，并允许低优先级的流量使用高优先级队列闲置的带宽，这样可以提高整个网络的效率。

（3）拥塞通告

当网络中发生拥塞时，由拥塞点向数据源发送指示来限制引起拥塞的流量，并在拥塞消失时通知其取消限制。拥塞通知提供了一种在二层网络对持续拥塞的流量端到端管理方法。

FCoE 采用增强型以太网作为物理网络传输架构，能够提供标准的光纤通道有效内容载荷，融合增强型以太网（CEE）可以避免类似 TCP/IP 协议的开销和数据包损失，如图 3-54 所示。



图 3-54　融合增强型以太网（CEE）

3.4 NAS 技术与应用

3.4.1 NAS 存储基础

NAS 和 SAN 最大的区别就在于 NAS 有文件操作和管理系统，而 SAN 却没有这样的系统功能，其功能仅仅停留在文件管理的下一层，即数据管理。SAN 和 NAS 并不是相互冲突的，是可以共存于一个系统网络中的，如图 3-55 所示。NAS 通过一个公共的接口实现空间的管理和资源共享，SAN 仅仅是为服务器存储数据提供一个专门的快速后方存储通道。

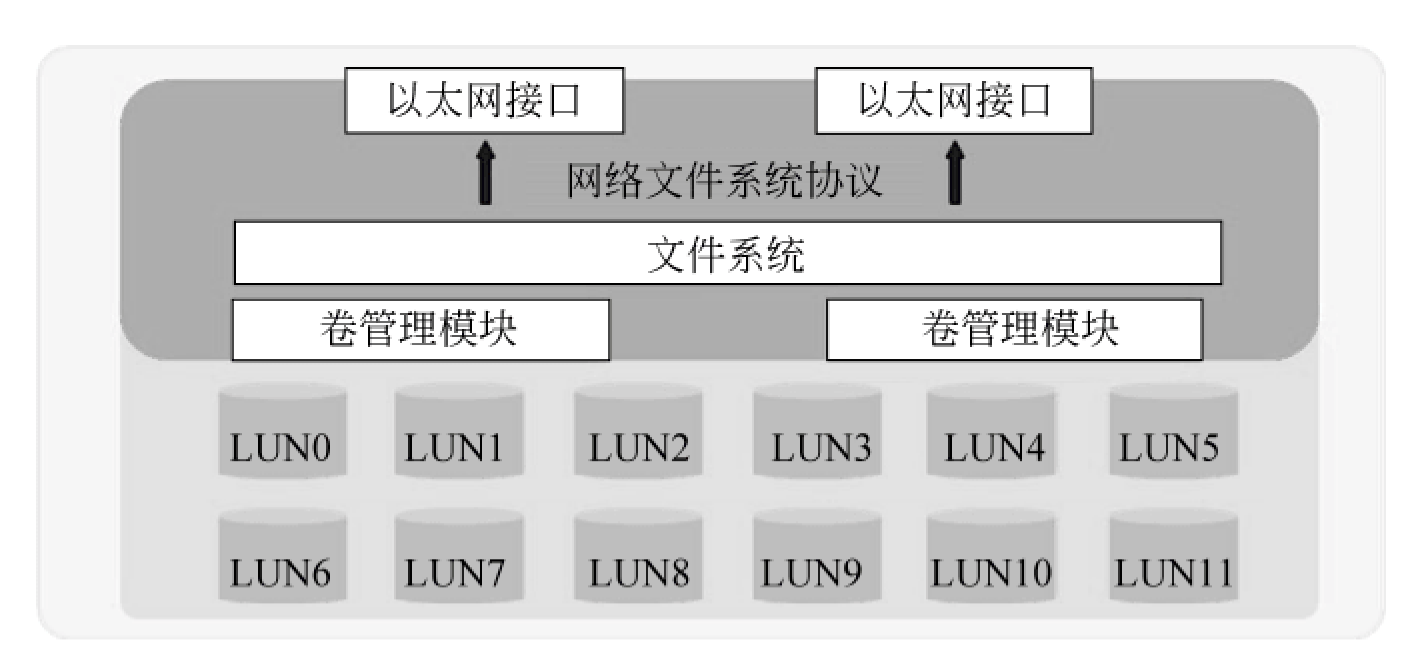


图 3-55　NAS 存储系统

FTP 文件服务不属于 NAS，FTP 只能将文件传输到本地的目录之后才能执行，而网络文件系统可以允许直接访问原始位置的文件，不需要将数据复制到本地再访问。

3.4.2 NAS 网络拓扑

NAS 可作为网络节点直接接入网络中，理论上 NAS 可支持各种网络技术，支持多种网络拓扑，但是以太网是目前最普遍的一种网络连接方式，我们主要讨论的是以以太网为网络基础的 NAS 环境，如图 3-56 所示。

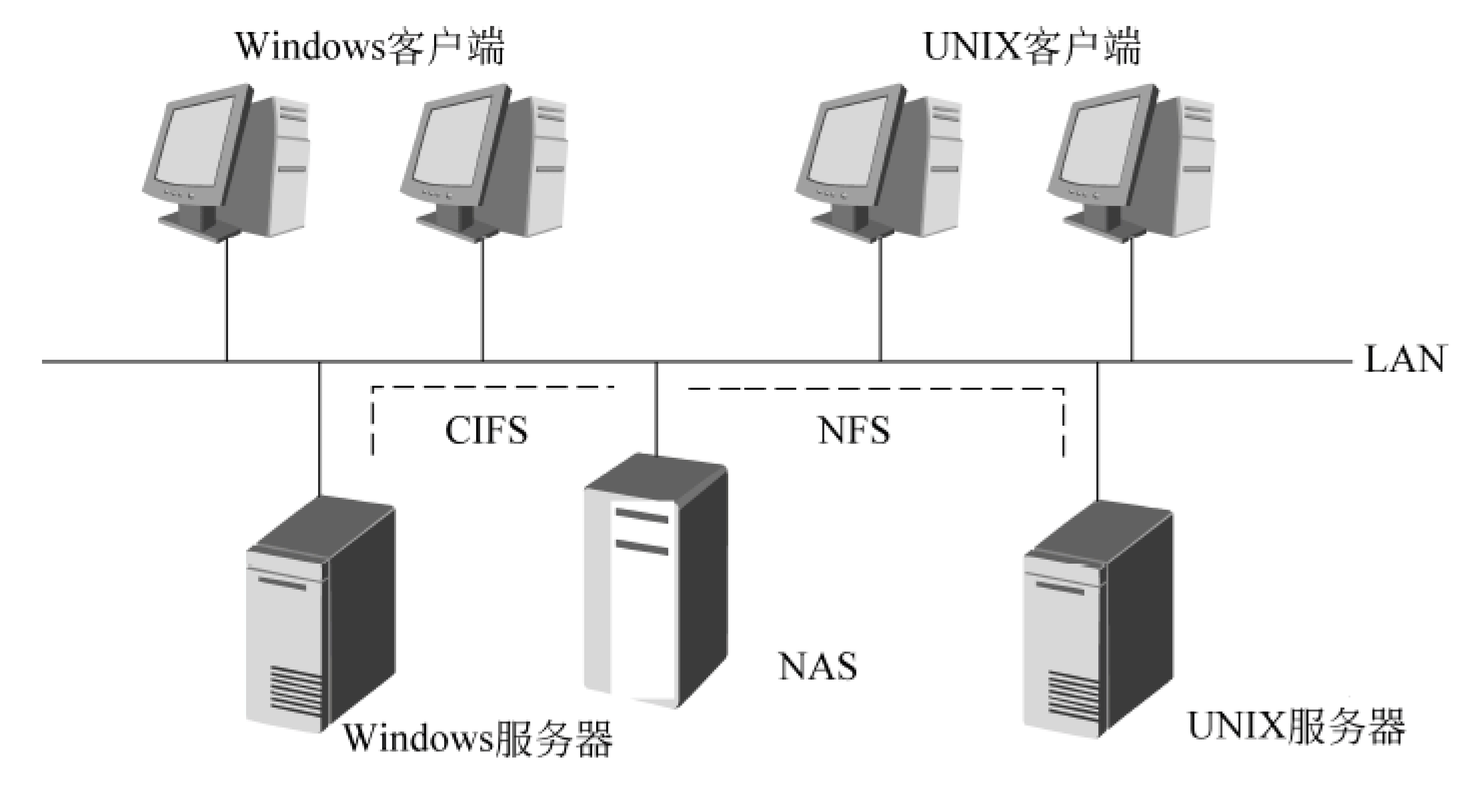


图 3-56　NAS 网络拓扑

NAS 本身能够支持多种协议(如 NFS、CIFS、FTP、HTTP 等),而且能够支持各种操作系统。通过任何一台工作站,采用 IE 或 Netscape 浏览器就可以对 NAS 设备进行直观方便的管理。

3.4.3　NAS 的发展及特点

传统的 DAS 存储系统尽管使用方便,但这种模式是直接将存储设备连接到服务器上。一方面,当存储容量增加时,这种方式很难扩展;另一方面,当服务器出现异常时,会使数据不可获得,容错性差;再者,存在着存储数据无法被其他服务器共享,扩充容量则需要关闭整个系统,远程管理不方便等诸多缺陷。于是便有了网络存储技术的出现,网络存储技术分为两类,即 SAN 和 NAS。在 SAN 环境中,存储设备通过网络与服务器相连,可以有更多的服务器访问存储设备提供的存储资源,存储设备提供数据块级别的服务。NAS 是一种向用户提供文件级服务的专用数据存储设备,直接连到网络上,不再挂接服务器后端,避免给服务器增加 I/O 负载,服务器只负责处理自身业务。

1. 集群技术的概念

一组相互独立的服务器在网络中表现为单一的系统,并以单一系统的模式加以管理。此单一系统为客户工作站提供高可靠性的服务。

2. 集群技术的特点

(1) 统一命名:大多数模式下,集群中所有的计算机拥有一个共同的名称,集群内任一系统上运行的服务可被所有的网络客户所使用。

(2) 高可靠性:集群必须可以协调管理各分离组件的错误和失败,集群内单一系统的失败由其他集群成员来弥补,对客户是不可见的。集群内部各节点服务器通过一内部局域网相互通信。当一台节点服务器发生故障时,这台服务器上所运行的应用程序将在另一节点服务器上被自动接管。当一个应用服务发生故障时,应用服务将被重新启动或被另一台服务器接管。当以上的任一故障发生时,客户都将能很快连接到新的应用服务上。

(3) 性能扩展:可透明地向集群中加入组件,提升系统的性能。

(4) 共享数据空间:一个集群包含多台(至少两台)拥有共享数据存储空间的服务器。任何一台服务器运行一个应用程序时,应用数据被存储在共享的数据空间内。每台服务器的操作系统和应用程序文件存储在其各自的本地存储空间上。

3.4.4　集群 NAS

集群 NAS 相较于传统 NAS,由于引擎采用了集群架构,带来了多方面的优势,引擎集群之间采用双向模式,可靠性更高。集群架构易扩展,增加引擎能线性提高性能。易扩展,新的引擎可直接加入集群,在线扩容,对业务不造成影响。易管理,对集群内的引擎节点可统一管理。集群 NAS 的优点如图 3-57 所示。

3.4.5　NAS 与文件服务器的对比

文件服务器的主要任务是为网络上的计算机提供多样化的服务,如在文件共享及处理、网页发布、FTP、电子邮件服务等方面有明显的优势,这主要得益于文件服务器通常是采取高性能的 CPU,与 NAS 相比它在数据备份、数据安全等方面并不占优势。

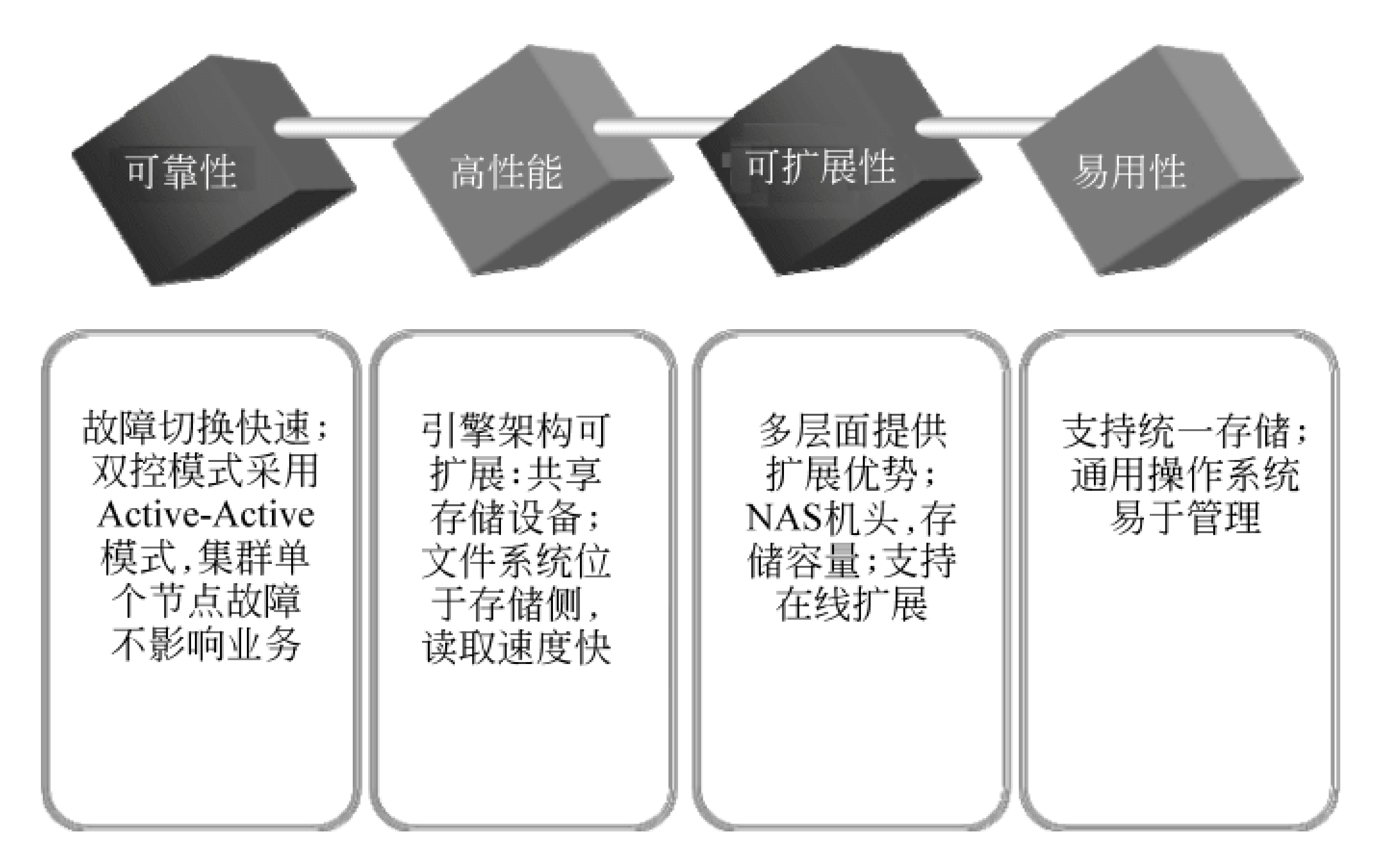


图 3-57　集群 NAS 的优点

通过对比不难看出，文件服务器相对于 NAS 综合功能特别是在文件处理能力方面更为强大，但文件服务器在数据的备份和恢复方面却远没有 NAS 的功能完善，系统稳定性也没有 NAS 的好，存储容量空间没有 NAS 的大及易扩展，同时在数据安全如数据容灾方面，NAS 更具优势。在这种状况下，两者无法相互替代，如图 3-58 所示。

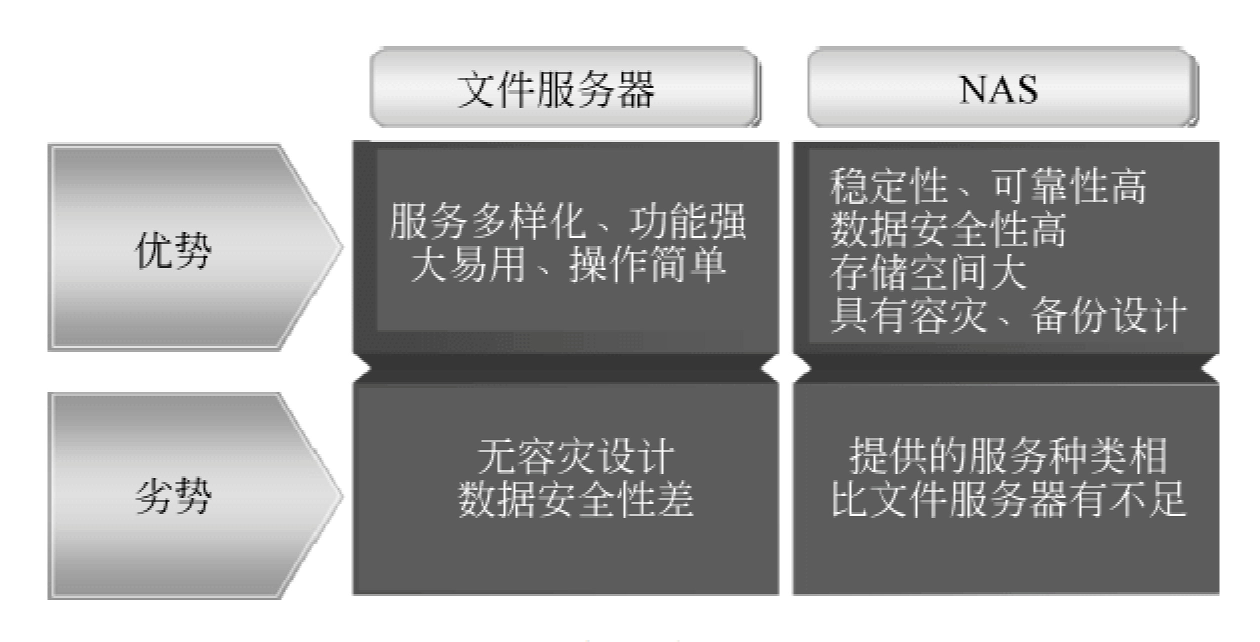


图 3-58　NAS 与文件服务器的对比

3.4.6　NAS 系统的组成与部件

1. NAS 系统架构

NAS 系统软件设计的基本要求是较高的稳定性和 I/O 吞吐率，并能满足数据共享、数据备份、安全配置、设备管理等要求。该结构分为五个模块：操作系统、存储管理器、文件系统、网络文件共享和 GUI 管理模块，如图 3-59 所示。

鉴于 Linux、FreeBSD 等免费的开放源码操作系统具有稳定、可靠、高效的优秀特性，现在大部分 NAS 设备都是基于此类操作系统开发的。

存储管理器的主要功能是磁盘和分区的管理，主要包括磁盘的监测与异常处理和逻辑卷的配置管理，一般应支持磁盘的热插拔、热替换等功能和 RAID0、RAID1、RAID5 类型的逻辑卷。存储管理器实现简化的、集中的存储管理功能，保证数据的完整性，并增强数据的可用性。

文件系统提供持久性存储和管理数据的手段，它必须是 32 位或以上并能支持多用户，

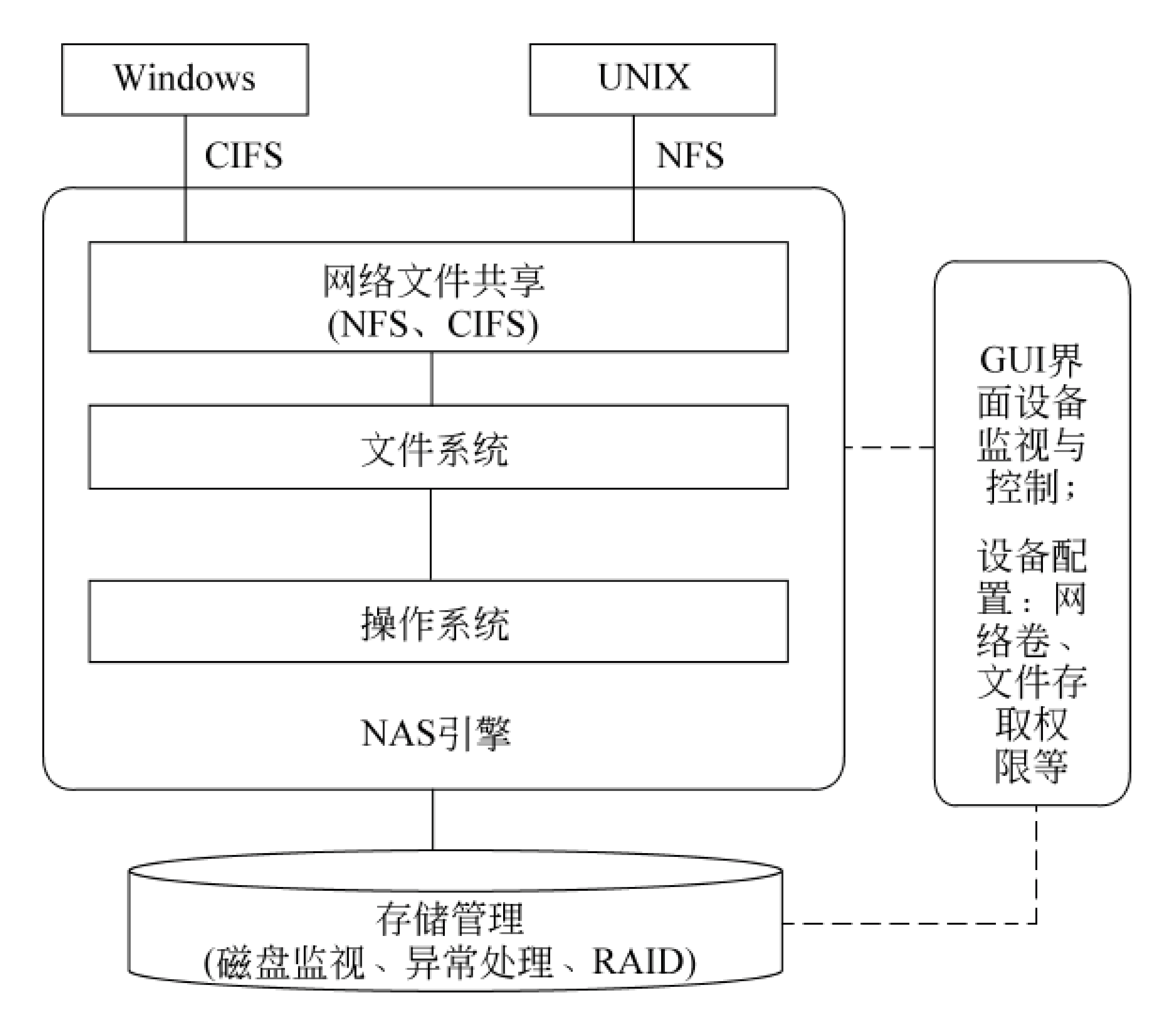


图 3-59　NAS 系统的架构

应具备日志文件系统功能以使系统在崩溃或掉电重启后能迅速恢复文件系统的一般性和完整性，进一步提供 NAS 的可用性。此外，文件系统还应具有快照功能。快照不仅能恢复被用户错误修改或删除的文件，而且能实现备份窗口为零的文件系统活备份。

网络文件共享一般支持以下一些文件传输和共享协议，如 FTP 和 HTTP 协议、UNIX 系统的 NFS、Windows 系统的 CIFS、Novell 系统的 NCP(Novell Core Protocol)、Apple 系统的 AFP(Appletalk File Protocol)等，因此 NAS 设备具有较好的协议独立性。

GUI 管理提供给系统管理员一个友好的界面，使之仅通过 Web 浏览器操作就能远程监视和管理 NAS 设备的系统参数，如网络配置、用户与组管理、卷以及文件共享权限等。

2. NAS 的组件

存储部分功能模块提供了真正的物理存储空间，主要技术是 RAID、SCSI、SAS、FC 等技术，如图 3-60 所示。

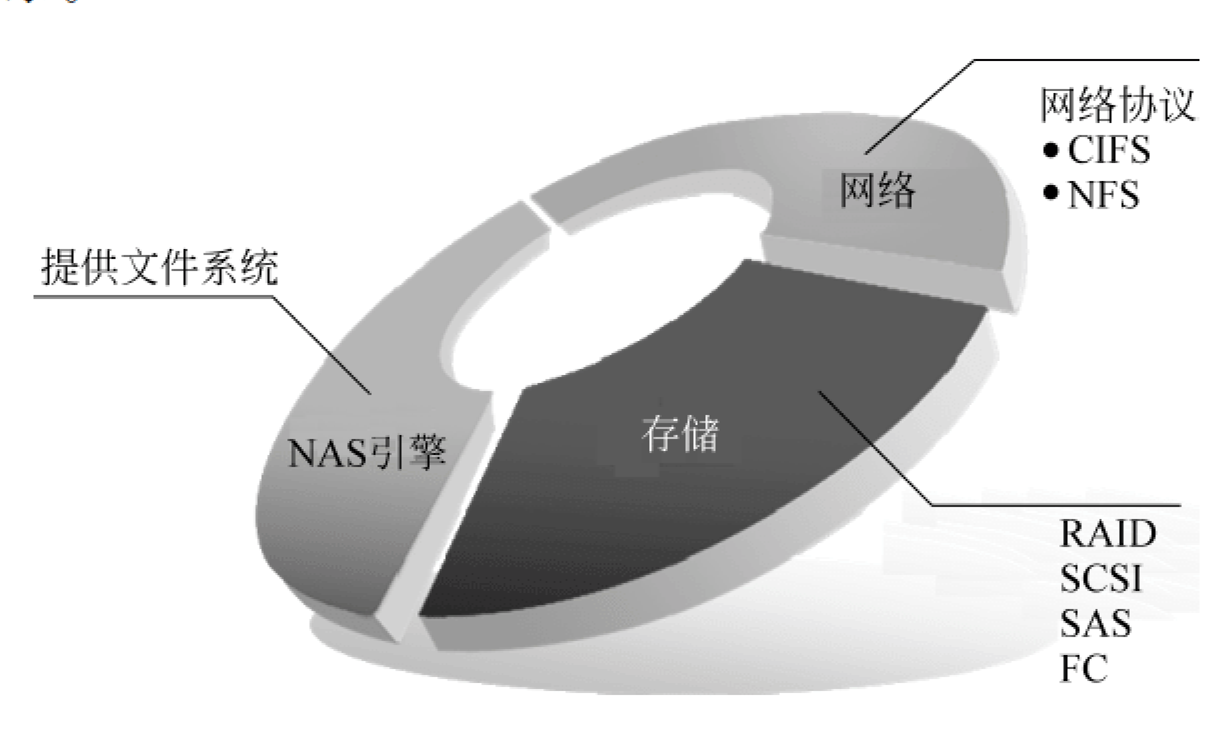


图 3-60　NAS 的组件

控制器部分指 NAS 引擎部分，这部分提供了 NAS 底层所使用的文件系统，以及承载文件系统、各种前端协议的操作系统。

网络部分提供了和用户交互的网络协议，主要包括 NFS 和 CIFS，用户最终通过这些协议访问存储空间。

3. NAS 引擎

NAS 引擎是 NAS 集群软件运行的硬件平台，通过软件的处理，将后端存储提供的卷以 NAS 方式提供给客户使用，但需要较好的 I/O 处理能力、网络带宽和可靠性，如图 3-61 所示。

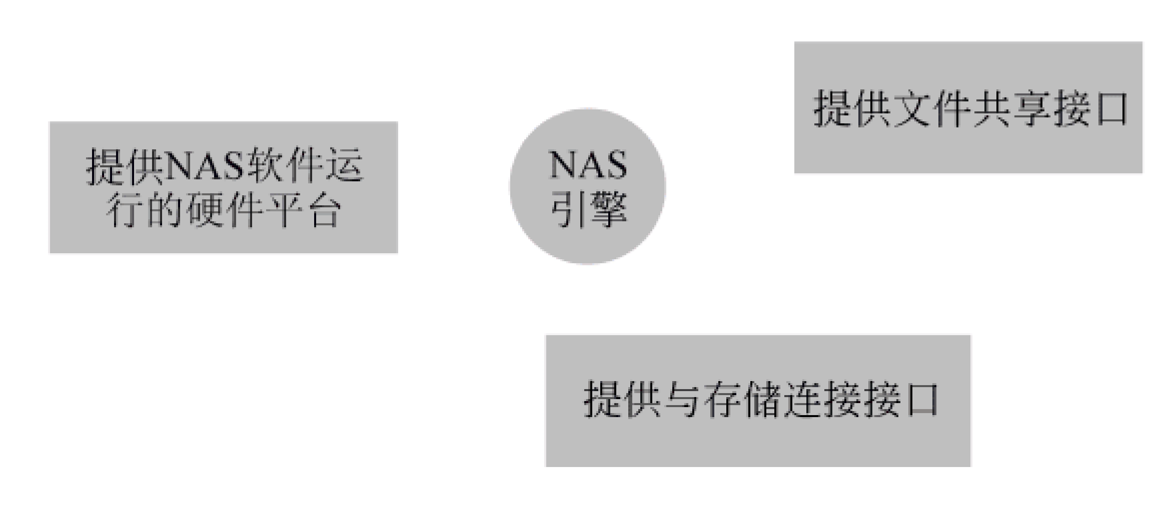


图 3-61　NAS 引擎

3.4.7　NAS 文件共享协议 CIFS 及 NFS

1. CIFS

CIFS 使用客户端/服务器模式，客户程序请求远在服务器上的服务器程序为它提供服务，服务器获得请求并返回响应，用在 Windows 系统共享文件的环境。当 NAS 系统对 Windows 系统服务器提供存储资源共享时，采用 CIFS 文件系统。

通用 Internet 文件系统(Common Internet File System，CIFS)是一个新提出的协议，它使程序可以访问远程 Internet 计算机上的文件并要求此计算机的服务。

2. NFS

网络文件系统(Network File System，NFS)是当前主流异构平台共享文件系统之一。主要应用在 UNIX 环境下。最早是由 SUN microsystem 开发，现在能够支持在不同类型的系统之间通过网络进行文件共享，广泛应用在 FreeBSD、SCO、Solaris 等异构操作系统平台，允许一个系统在网络上与他人共享目录和文件。通过使用 NFS，用户和程序可以像访问本地文件一样访问远端系统上的文件，使得每个计算机的节点能够像使用本地资源一样方便地使用网上资源。换言之，NFS 可用于不同类型计算机、操作系统、网络架构和传输协议运行环境中的网络文件远程访问和共享。

NFS 的工作原理是使用客户端/服务器架构，由一个客户端程序和服务器程序组成。服务器程序向其他计算机提供对文件系统的访问，其过程就称为“输出”。NFS 客户端程序对共享文件系统进行访问时，把它们从 NFS 服务器中“输送”出来。文件通常以“块”为单位进行传输。其尺寸是 8KB(虽然它可能会将操作分成更小尺寸的分片)。NFS 传输协议用于服务器和客户机之间文件访问和共享的通信，从而使客户机远程地访问保存在存储设备上的数据。

3. 网络模块

NAS 设备中所包含的标准文件系统可以对公用互联网文件系统(CIFS)或是网络文件系统(NFS)提供支持，也有可能同时支持两者。在许多情况下，它都使用标准的网络文件系

统来作为 NAS 专用文件系统的接口。大多数 NAS 设备需要用这种方式来管理其自身的存储资源。

NFS 是 UNIX 系统间实现磁盘文件共享的一种方法，支持应用程序在客户端通过网络存取位于服务器磁盘中数据的一种文件系统协议。其实它包括许多种协议，最简单的网络文件系统是网络逻辑磁盘，即客户端的文件系统通过网络操作位于远端的逻辑磁盘。现一般在 UNIX 主机之间采用 Sun 开发的 NFS(Sun)，它能够在所有 UNIX 系统之间实现文件数据的互访，逐渐成为主机间共享资源的一个标准。

CIFS 是由微软开发的，用于连接 Windows 客户机和服务器。经过 UNIX 服务器厂商的重新开发后，它可以用于连接 Windows 客户机和 UNIX 服务器，执行文件共享和打印等任务。它最早的由来是 NetBIOS，这是微软开发的在局域网内实现基于 Windows 名称资源共享的 API。之后，产生了基于 NetBIOS 的 NetBEUI 协议和 NBT(NetBIOS OVER TCP/IP)协议。NBT 协议进一步发展为 SMB(Server Message Block)和 CIFS(Common Internet File System，通用互联网文件系统)协议。其中，CIFS 用于 Windows 系统，而 SMB 广泛用于 UNIX 和 Linux，两者可以互通，SMB 协议还被称作 Lan Manager 协议。CIFS 支持与 SMB 的服务器通信而实现共享，微软操作系统家族和几乎所有 UNIX 服务器都支持 SMB 协议/SAMBA 软件包，如图 3-62 所示。

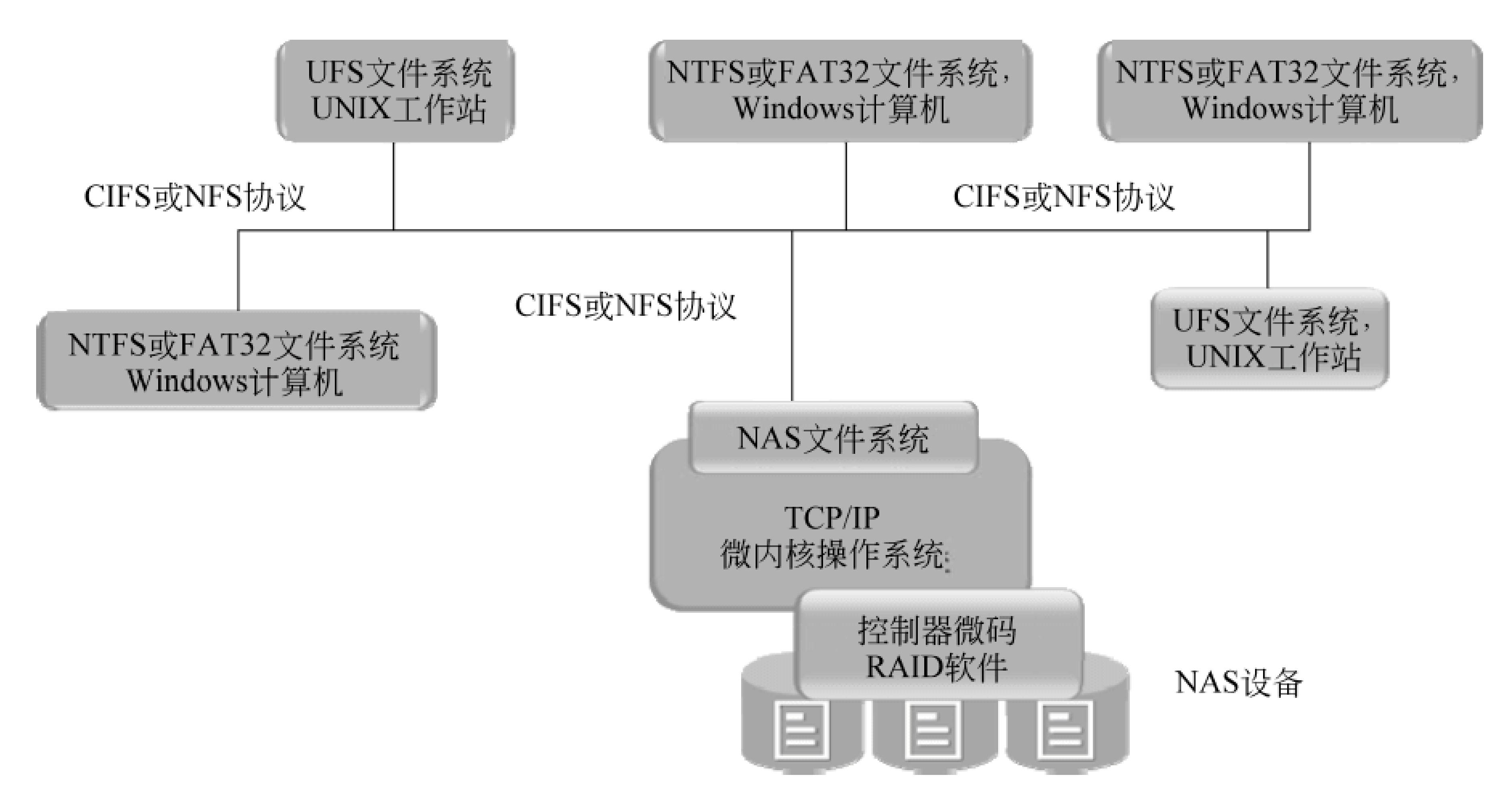


图 3-62　CIFS 或 NFS 协议

存储设备为 NAS 系统提供了真正的物理存储空间，数据通过 NAS 引擎处理以后，将数据保存到存储设备中。在主机上安装 iSCSI HBA 适配卡，从而实现主机与交换机之间、主机与存储设备之间的高效数据交换。iSCSI 层和 TCP/IP 协议栈的功能均由主机总线适配器(HBA)来完成，对主机 CPU 的占用最少。这种方式的数据传输性能最好，但是系统构建价格也最高。

4. CIFS 和 NFS 对比

(1) CIFS 面向网络连接的共享协议，对网络传输的可靠性要求高，常使用 TCP/IP；NFS 是独立于传输的，可使用 TCP 或 UDP。

(2) NFS 的缺点之一，是要求客户端必须安装专用软件；而 CIFS 集成在操作系统内

部，无须额外添加软件。

(3) NFS 属于无状态协议，而 CIFS 属于有状态协议；NFS 受故障影响小，可以自恢复交互过程，CIFS 不行；从传输效率上看，CIFS 优于 NFS，没有太多冗余信息传送。

(4) 两种协议都需要文件格式转换，NFS 保留了 UNIX 的文件格式特性，如所有人、组等；CIFS 则完全采用 Windows 的风格。

如果文件系统已经设置为 CIFS 共享，则此文件系统只能设置为只读的 NFS 共享。

如果文件系统已经设置为 NFS 共享，则此文件系统只能设置为只读的 CIFS 共享。

CIFS 和 NFS 的对比如表 3-4 所示。

表 3-4　CIFS 和 NFS 的对比

协议	传输协议	客户端要求	故障影响	效率	支持操作系统
CIFS	TCP/IP	操作系统集成，无须添加额外软件	大	高	Windows
NFS	TCP 或 UDP	需要额外的软件	小，可自恢复交互过程	低	UNIX

3.4.8　NAS 文件系统的 I/O 与性能

1. NAS 系统的 I/O 路径

在 NAS 系统中，NAS 引擎通过网络将后端的存储资源以文件夹的形式对外提供，当客户端在访问使用 NAS 存储资源的时候，一般由应用发起 I/O 操作，然后通过网络到达 NAS 引擎，引擎再对 I/O 操作进行处理，最后命令到达存储阵列，完成 I/O 操作。

2. FC-SAN 与 IP-SAN 解决方案的融合

统一存储解决方案典型 T 系列产品，可以同时对外提供 FC-SAN、IP-SAN 和 NAS 共享，如图 3-63 所示。

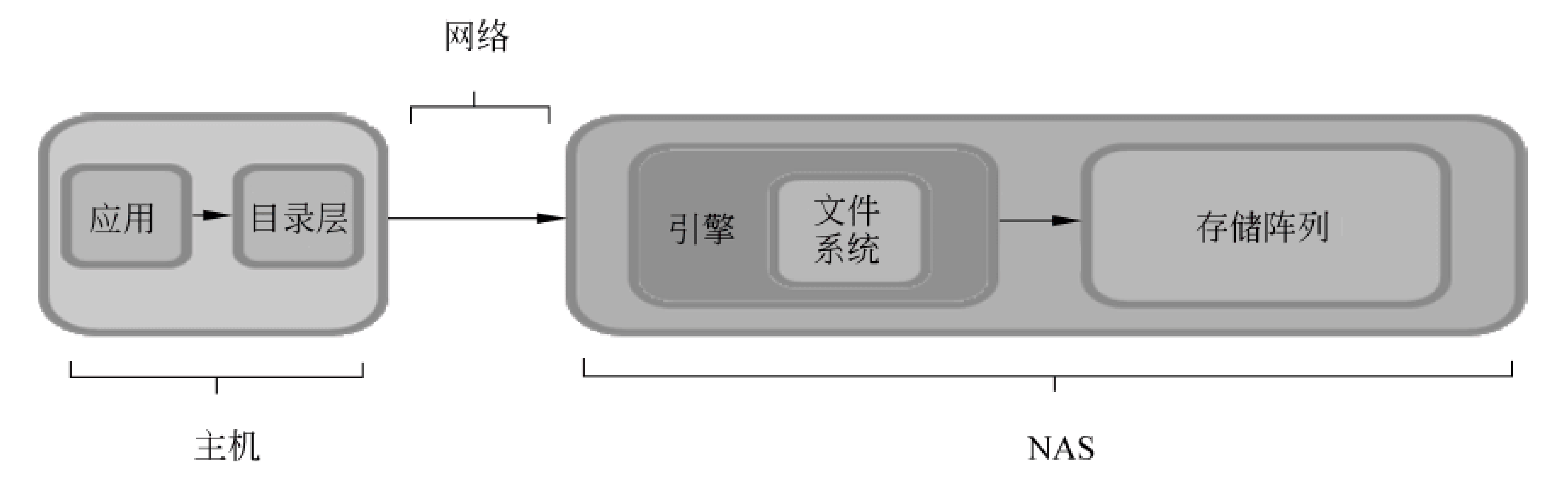


图 3-63　FC-SAN 与 IP-SAN 解决方案的融合

3. 受主机性能的影响

访问主机客户端本身配置较低，操作系统要花较多的时间来处理应用，如果再加上主机上运行的业务程序过多，则需要更多的时间去处理接收到的响应，那么就会影响到主机端应用对存储的操作，如图 3-64 所示。

4. 受网络性能的影响

跳转次数、大量的网络包跳转会增加延迟；重传、链路错误、缓冲区溢出和流量控制机制

都会导致重传。这会导致未到达指定目的地的报文被重新发送。在配置网络设备的速率参数、双工通信参数以及 NAS 头参数时要注意使它们彼此匹配。不恰当的配置会导致错误和重传，增加延迟，如图 3-65 所示。

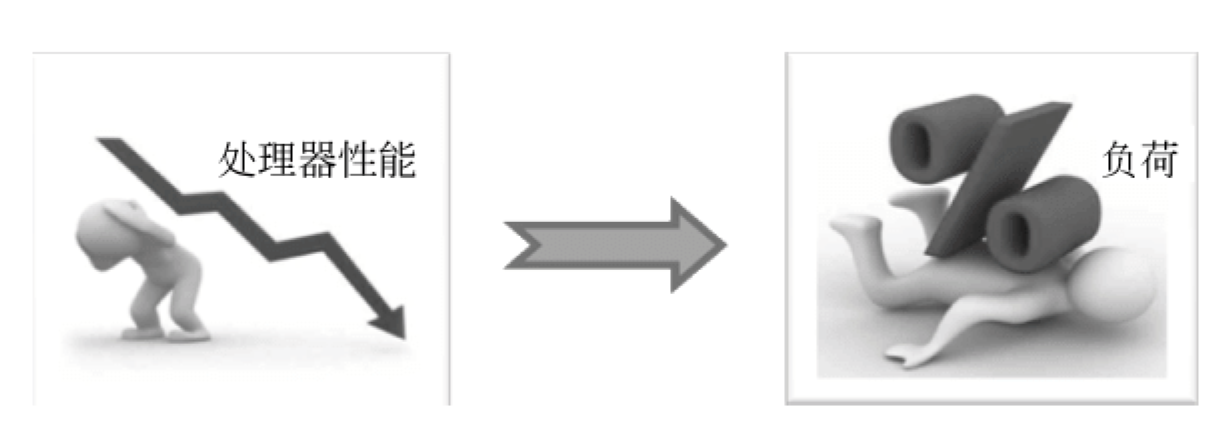


图 3-64　对主机性能影响

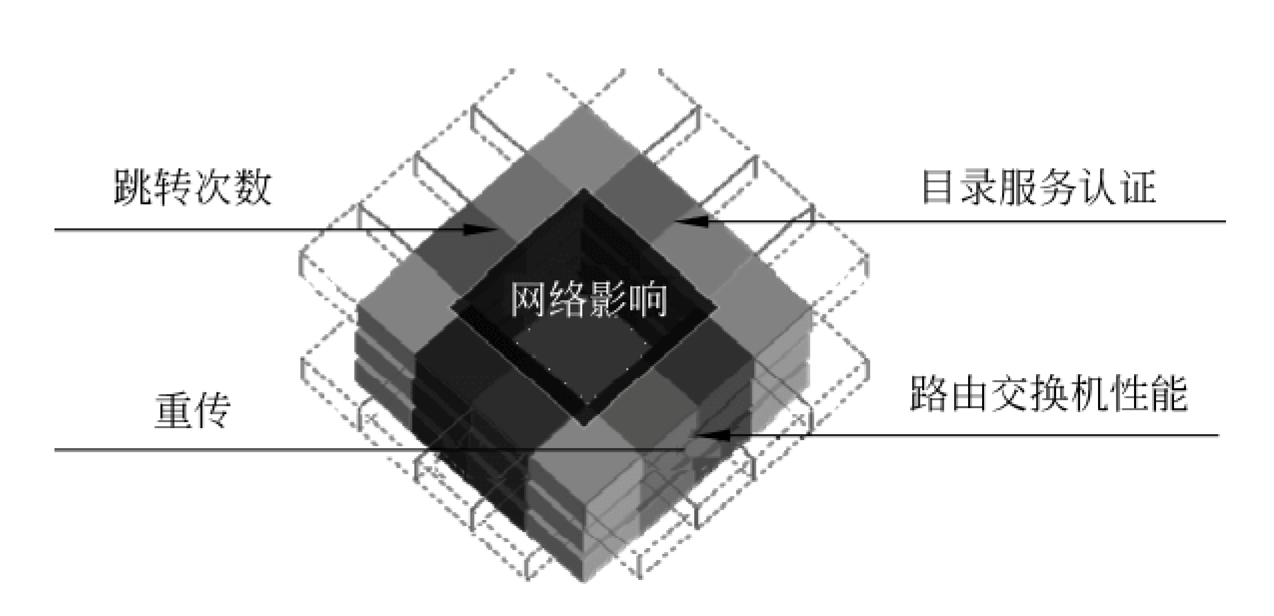


图 3-65　受网络性能的影响

（1）目录服务的认证

认证服务是网络上必需的服务，而且必须拥有充足的带宽和足够的资源来支持认证过程产生的负载，例如 LDAP、活动目录或 NIS。否则，大量的认证请求发向服务器会增加延迟。当然，只有当认证发生时才会增加延迟。

（2）路由交换机性能

在网络中，一个过载的设备所需要的响应时间总是比优化状态下使用的或低负载使用的设备所需要的响应时间要长。

5. 受 NAS 设备性能的影响

进行文件目录查找时，如果文件目录过大过深，在查找的时候会非常耗资源，对 NAS 性能造成一定的影响；设备过载，即长时间处于高负荷的状态，存储阵列降级、阵列降级状态，阵列内的磁盘会参与重构，一般重构数据量都比较大，也会直接影响阵列的性能，进而影响 NAS 的性能；而存储阵列设备性能低下会直接影响 NAS 的性能，因为 NAS 对数据的 I/O 操作最终的处理设备还是在阵列上进行的，如果阵列性能低下，整个 NAS 性能性能也就低下了，如图 3-66 所示。

3.4.9　NAS 的实现与应用

传统 NAS 非集群架构多为 Active-Standby 架构，NAS 引擎各自管理被分配的空间。

1. NAS 存储系统华为 N8500 产品

N8500 是华为一款集群化的中高端 NAS 存储系统，针对高效数据共享产品的需求，具有灵活的横向及纵向可扩展性，可用于金融、政府、石油天然气、健康和生命科学、制造业、

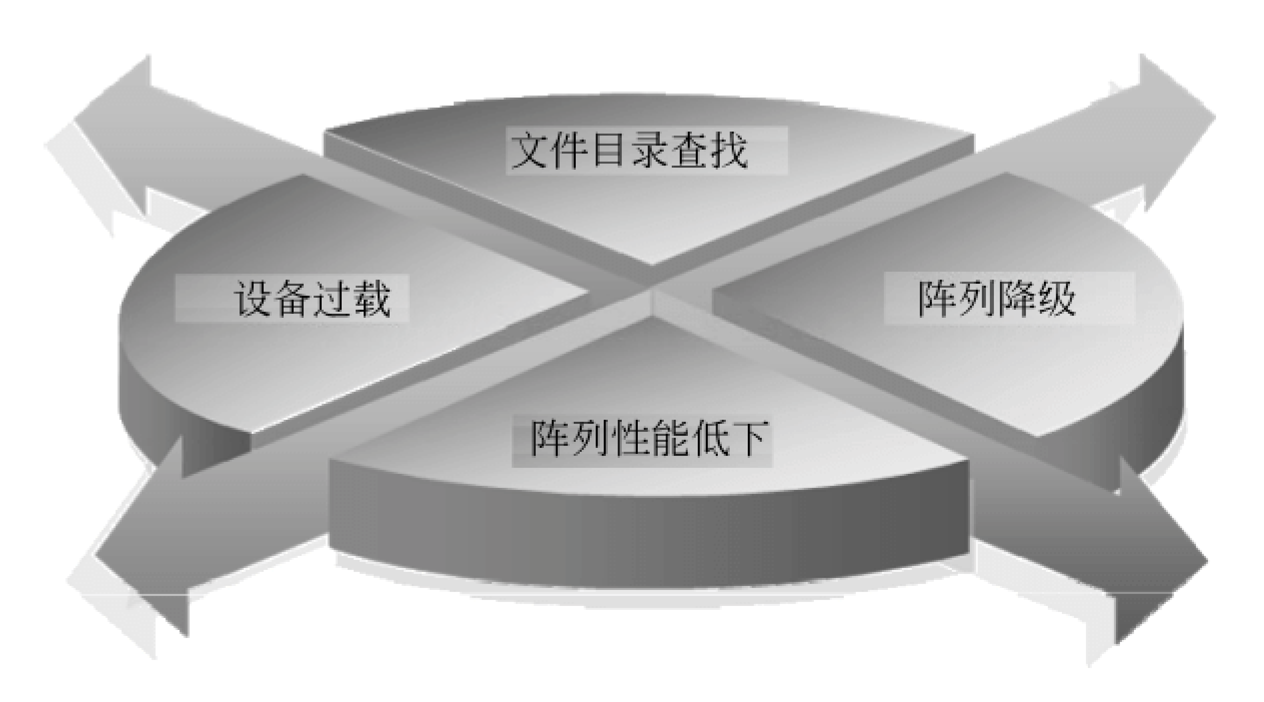


图 3-66　受网络性能的影响

E-Discovery 等行业。

N8500 的特点主要体现在高可靠性、高性能、支持动态分级存储、支持业务网口绑定、支持用户配额管理、支持文件系统快照、支持基于 LAN 的备份、支持文件系统多重镜像、支持文件系统在线扩容、支持 NFS 和 CIFS 协议的共享、支持 FTP 协议访问、支持域环境、支持存储单元后台格式化，如图 3-67 所示。



High Performance

- N8500 SPECsfs基准认证结果业界领先
- 性能线性展，按需购买

Scalability

- 领先的多节点全Active集群架构
- 最大支持24个引擎节点
- 系统最大支持15PB(PetaByte)存储容量

Efficiency

- 细粒度的动态分级存储功能
- 独有文件系统镜像功能

Convergence

- NFS、CIFS、iSCSI、FCP、FCoE、FTP、HTTP、NDMP等协议支持
- SAN和NAS统一管理

图 3-67　NAS 存储系统华为 N8500 产品及特点

2. N8500 架构的优势

如图 3-68 所示，N8500 采用集群架构，所有引擎均处于活动状态，NAS 引擎管理后端整个存储空间、文件系统 A 的数据，服务器群既可以通过 A 控制器访问，也能通过 B 控制器访问，文件系统 B 也是如此。通过多个控制器访问同一文件系统，提升并发处理效率，相当于若干个人同时干一个活，在 HPC、媒资等高性能存储市场中，效率可成倍提升。同时支持故障的自动切换，保障系统的持续运行。

（1）资源配置灵活

对于存储空间的管理，N8500 通过存储池方式对不同类型磁盘介质进行统一管理，可支持 SSD、SAS、NL-SAS、SATA 多种磁盘类型，可针对不同类型的应用，划分不同级别的资源空间，这样做可以实现按需分配，灵活调度，达到资源的最优配置。

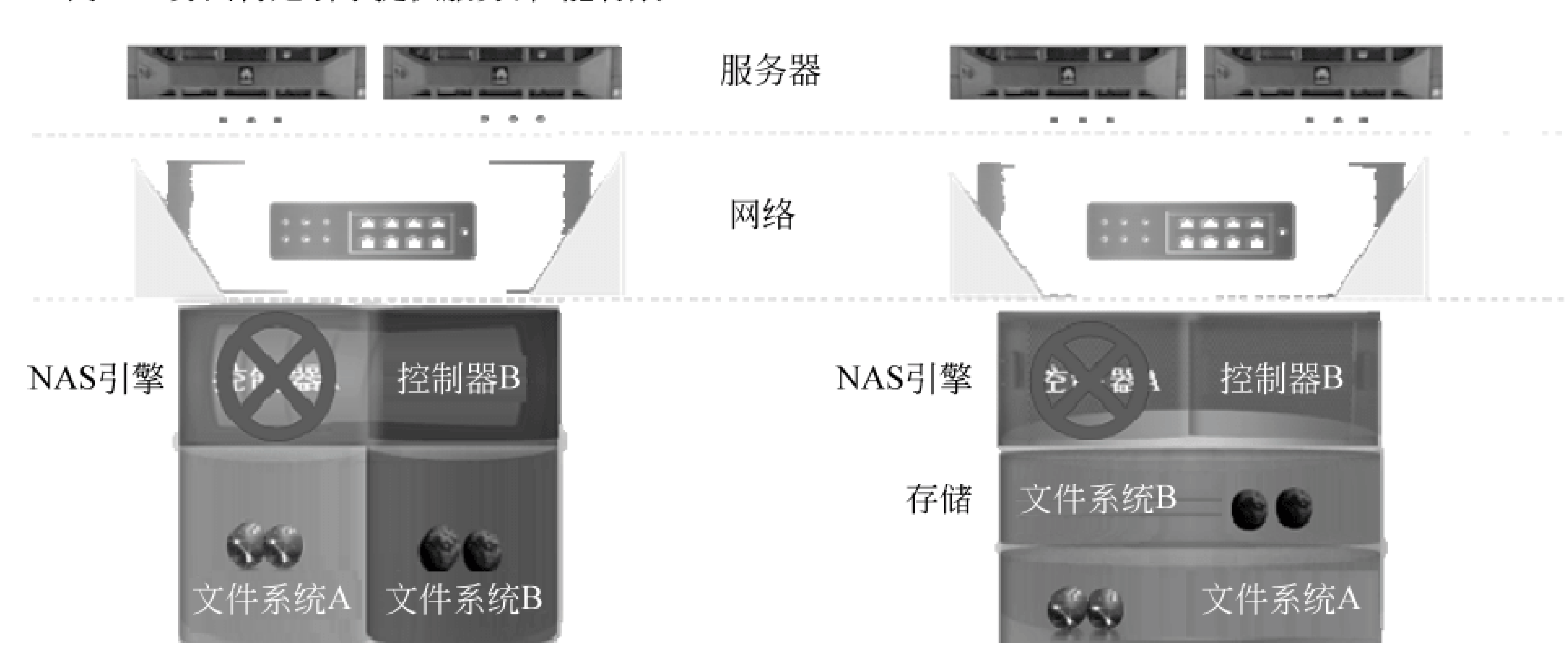


图 3-68　NAS 存储系统华为 N8500 构架的优势

(2) 智能迁移提升性能

智能迁移可提升性能并降低采购成本。大家都知道，数据可以根据其被使用的频度分为热点数据和非热点数据，N8500 通过 Smart Cache 技术和 DST 技术实现对热点数据的加速，以及对非热点数据的智能迁移。

DST 预测功能是通过对用户当前业务负载特点的分析，将负载高的数据预测分配到高性能层或者性能层，将负载非常低或者完全为 0 的数据预测分配到容量层，在满足用户当前业务性能需求且保证总价格最低的情况下，给出用户存储池中的所有业务数据在各层分布最合理的预测比例，以指导用户后续进行合理的配置。

SmartCache 技术通过对热点数据进行智能迁移，将热点数据镜像至 SSD 缓存池中，缓存池中始终保留最热点的数据，提升热点数据的访问性能，通常可提升 5 倍或以上。广泛适用于互联网，运营商彩铃彩信等应用场景。

而对于非热点数据的智能迁移，N8500 通过 DST 技术来实现，通常热点数据和非热点数据大约是 2∶8，将大量的非热点数据根据策略自动迁移至廉价的大容量 SATA 盘上，降低采购成本，通常可降低 60%的硬盘采购成本。相比业界的分级存储，DST 技术的迁移对象是针对文件进行迁移，热点识别更精准，迁移粒度更灵活。广泛适用于音视频点播、邮件系统、媒资库等存在大量"冷"数据的应用场景中，如图 3-69 所示。

(3) 本地数据保护

前面我们解决了存储的性能、扩展性、空间利用率，那么如何保证数据的安全呢，N8500 通过多种技术来保证数据的安全。

快照技术能有效解决数据的误删除，提升数据的安全性。

N8500 后端连接多套存储单元，数据同步镜像写入存储单元发生，一个存储单元发生故障时不影响系统的运行，提升了业务的可靠性。

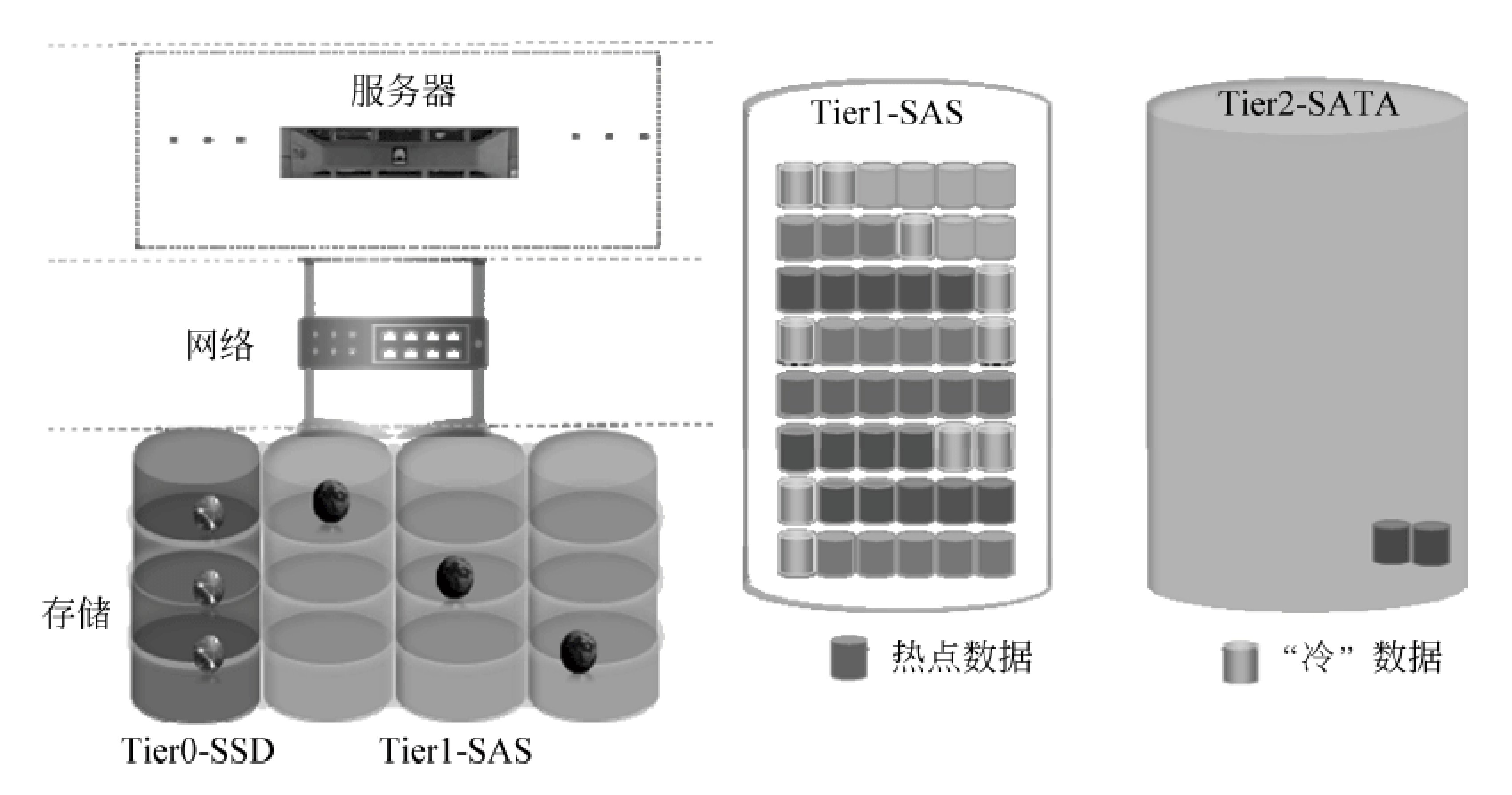


图 3-69 智能迁移提升性能

集成 NBU 的备份客户端，实现数据的高效备份。NBU 是赛门特克公司备份软件 Netbackup 的简称。如图 3-70 所示为本地数据的保护。



图 3-70 本地数据保护

(4) 异地数据容灾

有了本地的数据安全保护还不够，对于关键数据，有时还需要在异地保留一份数据。N8000 支持基于 SAN 空间和 NAS 空间的两个层级数据复制，进一步提升数据的安全性。远程复制是基于改变量的复制，占用网络资源少，缩短复制任务。生产站点数据一旦发生丢失，可通过灾备站点对数据进行恢复，如图 3-71 所示。

(5) 多协议融合

N8500 支持 FCP、FCoE、iSCSI 协议，相比在文件系统层面上再虚拟 SAN 空间的方式有更高的效率；支持基于文件的 NFS、CIFS、FTP、HTTP 访问协议，同时支持 NDMP、SNMP、Syslog 等其他协议。通过将不同协议融合至一套存储系统中，满足不同应用类型对存储的要求，简化 IT 基础架构，减少设备数量和采购成本，如图 3-72 所示。

NDMP 是由 Network Appliance Corporation 和 Legato System 公司合作开发的一种基于 NAS 的容灾备份技术。它提供了一个开放的协议，利用 NAS 设备进行数据的备份/

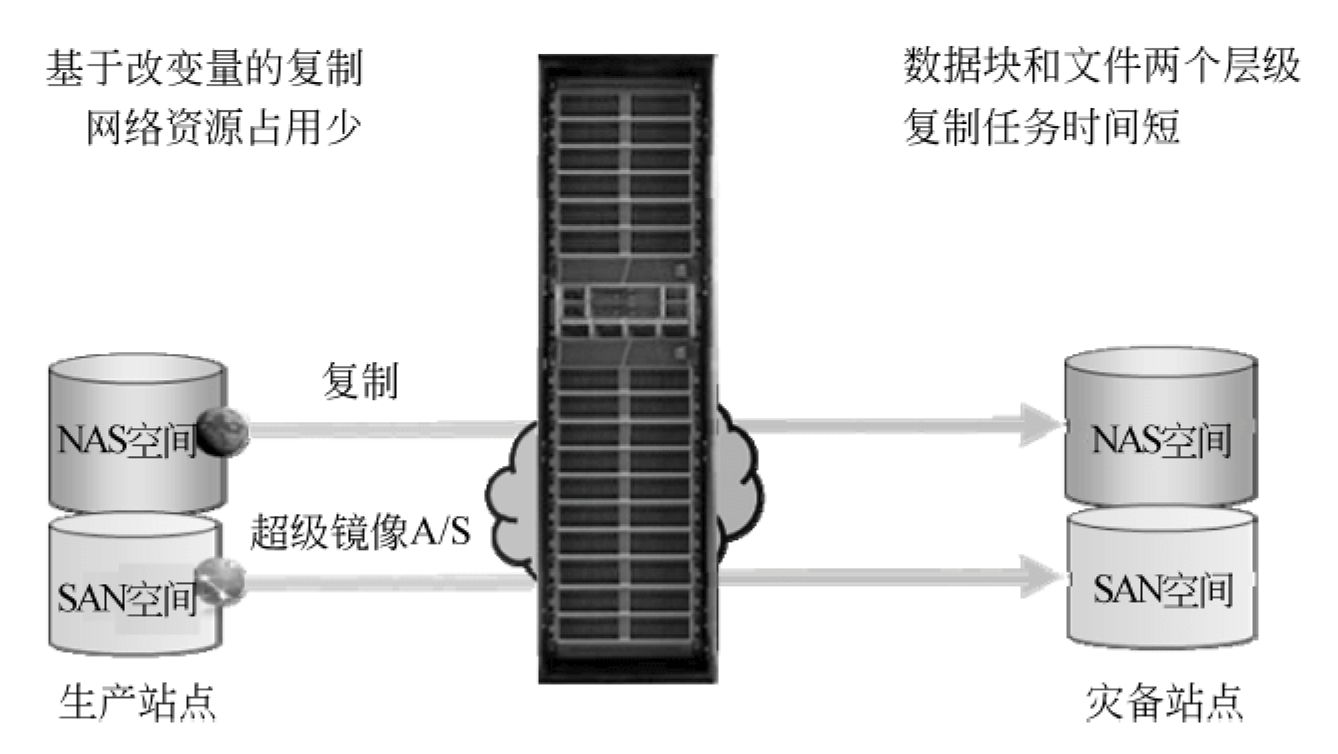


图 3-71　异地数据容灾

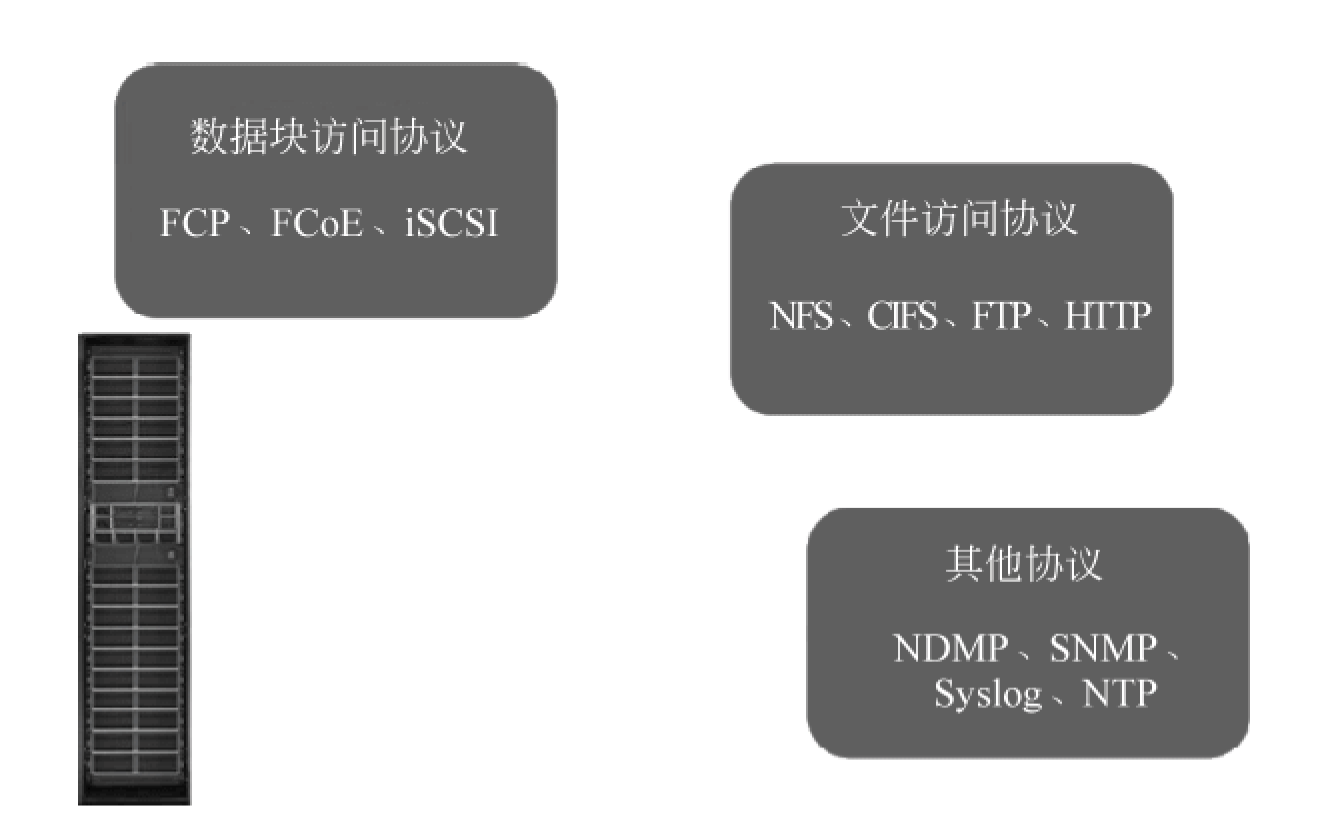


图 3-72　多协议融合

恢复，符合 NDMP 的备份应用程序可以通过它来控制任何运行 NDMP 服务器应用程序的 NDMP 主机的备份与恢复。

(6) 云备份场景——云备份服务商数据中心

云备份应用场景主要关注性能、扩展性和对海量文件的处理能力。华为 N8500 采用全 Active 集群架构，适合海量用户并发访问存储，同时随着用户数量的增大，数据容量不断增大，N8500 也能轻松扩容应对，能有效保护原有投资，如图 3-73 所示。

3. 存储整合场景——数字图书馆/档案馆

存储整合应用场景主要关注存储能提供丰富的接口和协议，提供丰富的软件特性，满足不同应用程序、不同服务等级对存储的不同需求，在数据中心放置一套存储就能满足各种需求。华为 N8500 提供多种接口模式，多种协议类型对外提供存储资源，能很好地满足存储整合应用场景的需求，如图 3-74 所示。

NAS 设备都分配有独立的 IP 地址，所以客户机通过充当数据网关的服务器可以对其进行存取访问，甚至不需要任何中间介质客户机也可以直接访问 NAS 设备，因此 NAS 还可以作为小型的邮箱服务器来使用，并且体积小巧、维护简单，再加上其售价相对低廉(与 DAS 和 SAN 相比)，因而受到一些中小企业的欢迎。而随着企业的壮大和云计算以及云存储的发展，数据量不断增大，NAS 的使用性能也似乎显得越来越力不从心。NAS 越来越不

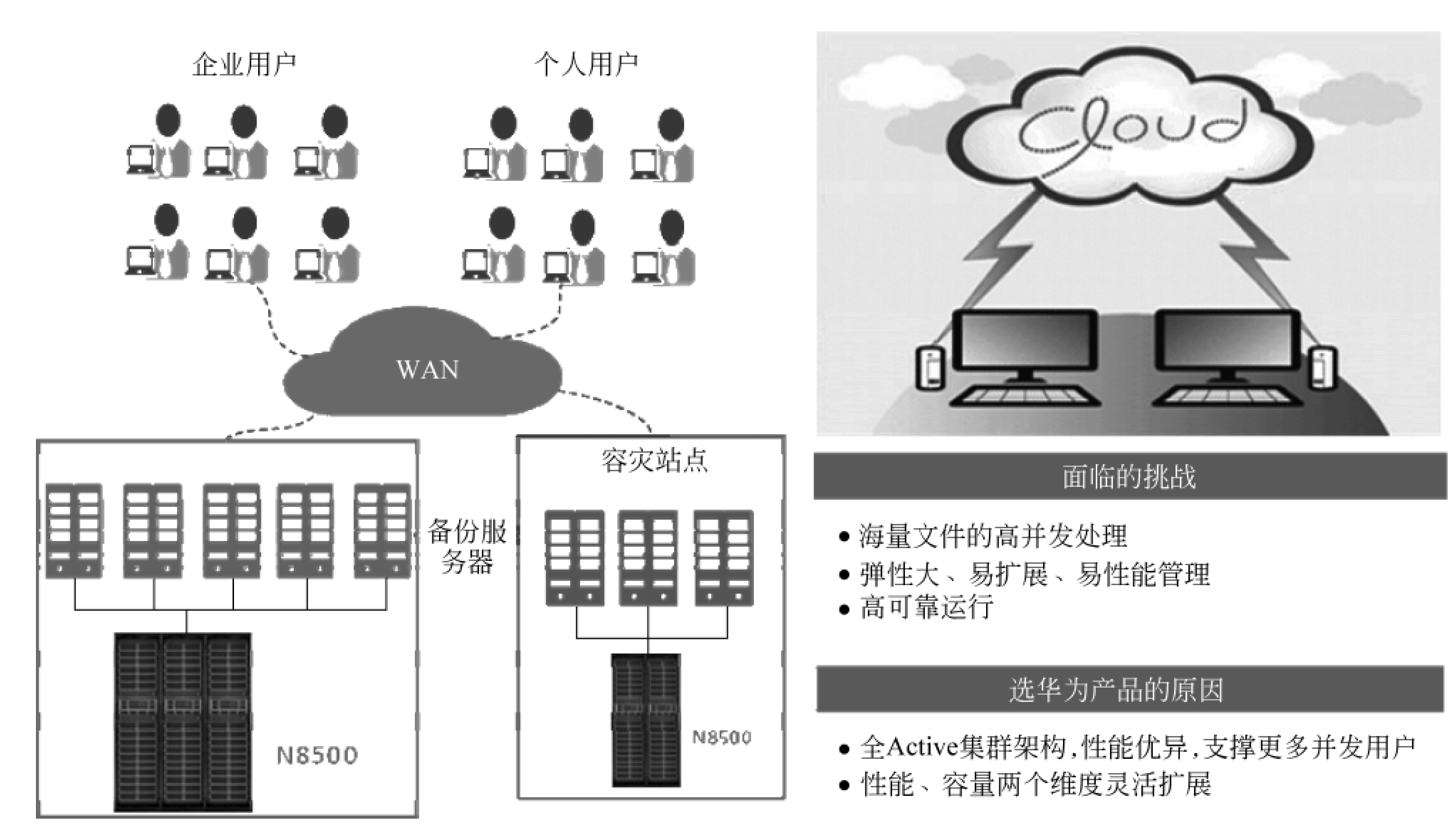


图 3-73　多协议的融合

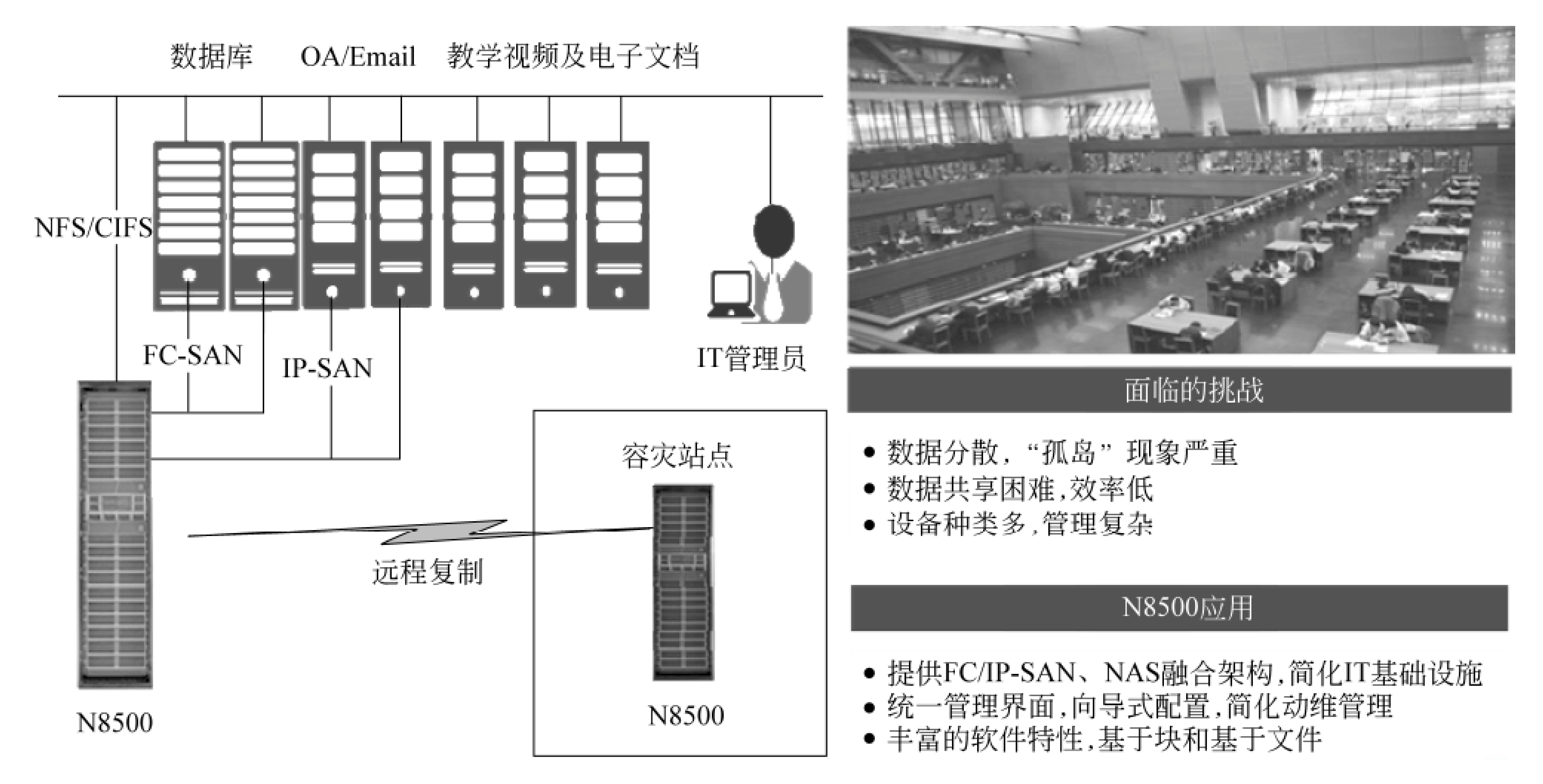


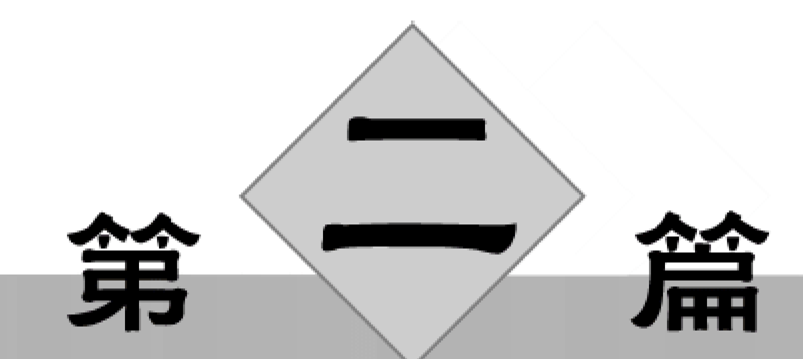
图 3-74　NAS 存储应用案例——数字图书馆/档案馆

适合作为企业的网络存储，但是这也并不意味着 NAS 就此将被淘汰。随着家庭多媒体的发展，人们也越来越重视家用计算机设备的发展。随着人们对高清片源的需求不断增大，越来越多的影视作品都是以全硬盘的方式出现，而一部全高清的带 DTS 音轨的电影容量可以达到十几甚至几十千兆字节(GB)，并且为了满足在卧室和客厅能够同时欣赏的需求，低成本网络存储就成了家庭存储方案的最好选择。不仅如此，我们的其他家电也越来越智能化，像数码相机、数码摄像机、电视机等都可以接入局域网中，因此网络存储也成了绝佳的多媒体仓库。于是 NAS 又走进了人们的视线。

任务拓展

1. 简述存储阵列系统的架构。
2. 简述 NAS 与 SAN 的区别。
3. 了解各种存储技术的案例与应用。





数据灾备概述

第 4 章　RAID 技术与应用

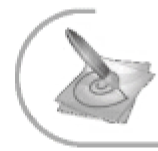
第 5 章　虚拟磁带库技术

第 6 章　数据灾备与恢复技术

第 7 章　虚拟化技术



第 4 章　RAID 技术与应用



任务目标

- 了解 RAID 技术的基本概念；
- 了解 RAID 的不同级别；
- 了解 RAID 不同级别的应用方向。

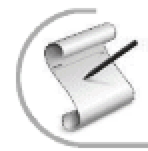


项目背景

20 世纪 80 年代后期，计算机在商业处理领域的迅速普及促进了新应用程序和数据库的增长，进而显著提高了人们对于存储容量和性能的需求。在当时，数据通常存储在单个被称为 SLED(SingleLargeExpensice Drive)的大容量、昂贵的磁盘驱动器中。单一磁盘无法满足人们对性能的需求，因为它们同时只能提供有限数量的 I/O。

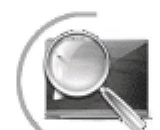
在大型数据中心的存储架构中，硬盘的数量少则几百，多则上万。在机械磨损等环境因素的影响下，磁盘驱动器非常容易发生故障。硬盘故障通常会导致数据的丢失。存储架构中硬盘的数量越多，发生硬盘故障的概率就越大。

如何解决单一硬盘性能不足和由硬盘正常损耗导致的数据安全问题，正式摆在人们面前。



项目描述

聚比特科技有限公司准备发展图像处理业务，但突然发现，图像处理工作需要确保拥有大量的不断增长的高清图像数据，而这类数据十分巨大，公司现有服务器硬盘根本无法承载，从而导致新业务工作无法开展，各级领导对此事十分重视，责成技术部门尽快解决。



项目分析

聚比特科技有限公司技术部门对业务进行了详细分析，确认两种解决方案，一种是通过增加服务器硬盘数量，将图像文件分别存储到各颗硬盘上。另一种是购买磁盘阵列，通过 RAID 技术将多个硬盘整合到一起，组成一个大的存储系统，将数据统一存放到磁盘阵列中。

通过比对发现，第一种方案成本较低，但由于文件分开存放，导致业务系统调用困难。而且，当硬盘数量越来越大时，单个硬盘故障的概率也随之增加，一旦某个硬盘发生故障，就会导致部分文件的丢失。

第二种方案成本相对较高，但其是通过 RAID 技术把所有硬盘整合到一起，方便了业务

系统对数据的调用，并且对硬盘故障有冗余保护，如果发生单个硬盘故障，不会导致数据丢失，对数据的保护能力大大增强了。

由于第二种方案在业务系统和文件数据方面的巨大优势，大大超过了其对应的成本增加，所以技术部门决定采用第二种方案来为公司的新业务提供大容量存储支持。



项目实现

聚比特科技有限公司根据技术部门的反馈，采购了一套由磁盘阵列构成的存储系统，通过 RAID 技术，将大量的硬盘整合到一起组合成一个文件系统，提供给图像处理系统使用，使得其图像处理业务顺利发展起来。

4.1 RAID 技术介绍

4.1.1 RAID 简介

磁盘阵列（Redundant Arrays of Independent Disks，RAID），有“独立磁盘构成的具有冗余能力的阵列”之意。

RAID 技术作为高性能、高可靠的存储技术，应用非常广泛。RAID 主要利用数据条带、数据校验技术来获取高性能、高可靠性和高扩展性，根据运用或组合运用这三种技术的策略和架构，可以把 RAID 分为不同的等级，以满足不同数据应用的需求。D. A. Patterson 等的论文中定义了 RAID1～RAID5 原始的 RAID 等级，1988 年以来又扩展了 RAID0 和 RAID6。近年来，存储厂商不断推出诸如 RAID7、RAID10/01、RAID50、RAID53、RAID100 等 RAID 等级，但这些并无统一的标准。目前业界公认的标准是 RAID0～RAID5，除 RAID2 外的 5 个等级被指定为工业标准，而在实际应用领域中使用最多的 RAID 等级是 RAID0、RAID1、RAID3、RAID5、RAID6、RAID01 和 RAID10。

4.1.2 RAID 的实现技术分类

从实现角度看，RAID 主要分为软 RAID、硬 RAID 以及软硬混合 RAID 三种。软 RAID 的所有功能均由操作系统和 CPU 来完成，没有独立的 RAID 控制/处理芯片和 I/O 处理芯片，效率自然最低。硬 RAID 配备了专门的 RAID 控制/处理芯片和 I/O 处理芯片以及阵列缓冲，不占用 CPU 资源，但成本很高。软硬混合 RAID 具备 RAID 控制/处理芯片，但缺乏 I/O 处理芯片，需要 CPU 和驱动程序来完成，性能和成本在软 RAID 和硬 RAID 之间。

4.1.3 RAID 的基本原理

简单来说，RAID 是由多个独立的高性能磁盘驱动器组成的磁盘子系统，从而提供比单个磁盘更高的存储性能和数据冗余的技术。RAID 是一类多磁盘管理技术，其向主机环境提供了成本适中、数据可靠性高的高性能存储。通常对 RAID 的定义是：一种磁盘阵列，部分物理存储空间用来记录保存在剩余空间上的用户数据的冗余信息。当其中某一个磁盘或访问路径发生故障时，冗余信息可用来重建用户数据。磁盘条带化虽然与 RAID 定义不符，

通常还是称为 RAID(即 RAID0)。

RAID 的初衷是为大型服务器提供高端的存储功能和冗余的数据安全。在整个系统中,RAID 被看作是由两个或更多磁盘组成的存储空间,通过并发地在多个磁盘上读写数据来提高存储系统的 I/O 性能。大多数 RAID 等级具有完备的数据校验、纠正措施,从而提高系统的容错性甚至镜像方式,大大增强系统的可靠性,Redundant 也由此而来。

这里要提一下 JBOD(Just a Bunch of Disks)。最初 JBOD 用来表示一个没有控制软件提供协调控制的磁盘集合,这是 RAID 区别于 JBOD 的主要因素。目前 JBOD 常指磁盘柜,而不论其是否提供 RAID 功能。

RAID 的两个关键目标是提高数据可靠性和 I/O 性能。磁盘阵列中,数据分散在多个磁盘中,然而对于计算机系统来说,就像一个单独的磁盘。通过把相同数据同时写入到多块磁盘(典型地如镜像),或者将计算的校验数据写入阵列中来获得冗余能力,当单块磁盘出现故障时可以保证不会导致数据丢失。有些 RAID 等级允许更多的磁盘同时发生故障,比如 RAID6,可以是两块磁盘同时损坏。在这样的冗余机制下,可以用新磁盘替换故障磁盘,RAID 会自动根据剩余磁盘中的数据和校验数据重建丢失的数据,保证数据的一致性和完整性。数据分散保存在 RAID 中的多个不同磁盘上,并发数据读写要大大优于单个磁盘,因此可以获得更高的聚合 I/O 带宽。当然,磁盘阵列会减少全体磁盘的总可用存储空间,并通过牺牲空间换取更高的可靠性和性能。比如,RAID1 存储空间利用率仅有 50%,RAID5 会损失其中一个磁盘的存储容量,空间利用率为$(n-1)/n$。

磁盘阵列可以在部分磁盘(单块或多块,根据实现而论)损坏的情况下仍能保证系统不中断地连续运行。在重建故障磁盘数据至新磁盘的过程中,系统可以继续正常运行,但是性能方面会有一定程度上的降低。一些磁盘阵列在添加或删除磁盘时必须停机,而有些则支持热交换(Hot Swapping),允许不停机下替换磁盘驱动器。这种高端磁盘阵列主要用于要求高可能性的应用系统,系统不能停机或尽可能少停机。一般来说,RAID 不可作为数据备份的替代方案,它对非磁盘故障等造成的数据丢失无能为力,比如病毒、人为破坏、意外删除等情形。此时的数据丢失是相对操作系统、文件系统、卷管理器或者应用系统来说的,对于 RAID 系统来说,数据都是完好的,没有发生丢失。所以,数据备份、灾备等数据保护措施是非常必要的,与 RAID 相辅相成,保护数据在不同层次的安全性,防止发生数据丢失。

4.1.4 RAID 的关键技术

RAID 中主要有三个关键概念和技术:镜像(Mirroring)、数据条带(Data Stripping)和数据校验(Data Parity)。

1. 镜像

镜像是一种冗余技术,为磁盘提供保护功能,防止磁盘因发生故障而造成数据丢失。对于 RAID 而言,采用镜像技术典型地将会同时在阵列中产生两个完全相同的数据副本,分布在两个不同的磁盘驱动器组上。镜像提供了完全的数据冗余能力,当一个数据副本失效不可用时,外部系统仍可正常访问另一个数据副本,不会对应用系统的运行和性能产生影响。而且,镜像不需要额外的计算和校验,故障修复非常快,直接复制即可。镜像技术可以从多个副本进行并发读取数据,提供更高的读 I/O 性能,但不能并行写数据,写多个副本会导致一定的 I/O 性能降低。

镜像技术提供了非常高的数据安全性，其代价也是非常昂贵的，需要至少双倍的存储空间。高成本限制了镜像的广泛应用，主要应用于至关重要的数据保护，这种场合下数据丢失会造成巨大的损失。另外，镜像通过“拆分”能获得特定时间点的数据快照，从而可以实现一种备份窗口几乎为零的数据备份技术。

2. 数据条带

磁盘存储的性能瓶颈在于磁头寻道定位，它是一种慢速机械运动，无法与高速的CPU匹配。再者，单个磁盘驱动器性能存在物理极限，I/O性能非常有限。RAID由多块磁盘组成，数据条带技术将数据以块的方式分布存储在多个磁盘中，从而可以对数据进行并发处理。这样写入和读取数据就可以在多个磁盘上同时进行，并发产生非常高的聚合I/O，有效提高了整体I/O性能，而且具有良好的线性扩展性。这对大容量数据尤其显著，如果不分块，数据只能按顺序存储在磁盘阵列的磁盘上，需要时再按顺序读取。而通过条带技术，可获得数倍于顺序访问的性能提升。

数据条带技术的分块大小选择非常关键。条带粒度可以是一个字节甚至至几千字节(KB)大小，分块越小，并行处理能力就越强，数据存取速度就越高，但同时就会增加块存取的随机性和块寻址时间。实际应用中，要根据数据特征和需求来选择合适的分块大小，在数据存取随机性和并发处理能力之间进行平衡，以争取尽可能高的整体性能。

数据条带是基于提高I/O性能而提出的，也就是说它只关注性能，而对数据可靠性、可用性没有任何改善。实际上，其中任何一个数据条带损坏都会导致所有数据不可用，采用数据条带技术反而增加了数据发生丢失的概率。

3. 数据校验

镜像具有高安全性、高读性能，但冗余开销太昂贵。数据条带通过并发性来大幅提高性能，然而对数据安全性、可靠性未作考虑。数据校验是一种冗余技术，它用校验数据来提供数据的安全，可以检测数据错误，并在能力允许的前提下进行数据重构。相对于镜像，数据校验大幅缩减了冗余开销，用较小的代价换取了极佳的数据完整性和可靠性。数据条带技术提供高性能，数据校验提供数据安全性，RAID不同等级往往同时结合使用这两种技术。

采用数据校验时，RAID要在写入数据的同时进行校验计算，并将得到的校验数据存储在RAID成员磁盘中。校验数据可以集中保存在某个磁盘或分散存储在多个不同磁盘中，甚至校验数据也可以分块，不同RAID等级实现各不相同。当其中一部分数据出错时，就可以对剩余数据和校验数据进行反校验计算来重建丢失的数据。校验技术相对于镜像技术的优势在于节省了大量开销，但由于每次数据读写都要进行大量的校验运算，对计算机的运算速度要求很高，必须使用硬件RAID控制器。在数据重建恢复方面，检验技术比镜像技术复杂得多且慢得多。

海明校验码和异或校验是两种最为常用的数据校验算法。海明校验码是由理查德·海明提出的，不仅能检测错误，还能给出错误位置并自动纠正。海明校验的基本思想是：将有效信息按照某种规律分成若干组，对每一个组作奇偶测试并安排一个校验位，从而能提供多位检错信息，以定位错误点并纠正。可见海明校验实质上是一种多重奇偶校验。异或校验通过异或逻辑运算产生，将一个有效信息与一个给定的初始值进行异或运算，会得到校验信息。如果有效信息出现错误，通过校验信息与初始值的异或运算能还原正确的有效信息。

4.1.5　RAID 的优势

RAID 思想从提出后就广泛被业界所接纳，存储工业界投入了大量的时间和财力来研究和开发相关产品。而且，随着处理器、内存、计算机接口等技术的不断发展，RAID 不断地发展和革新，在计算机存储领域得到了广泛的应用，从高端系统逐渐延伸到普通的中低端系统。RAID 技术如此流行，源于其具有显著的特征和优势，基本可以满足大部分的数据存储需求。总体来说，RAID 主要优势有如下几点。

1. 大容量

这是 RAID 的一个显然优势，它扩大了磁盘的容量，由多个磁盘组成的 RAID 系统具有海量的存储空间。现在单个磁盘的容量就可以到 1TB 以上，这样 RAID 的存储容量就可以达到 PB 级，大多数的存储需求都可以满足。一般来说，RAID 可用容量要小于所有成员磁盘的总容量。不同等级的 RAID 算法需要一定的冗余开销，具体容量开销与采用算法相关。如果已知 RAID 算法和容量，可以计算出 RAID 的可用容量。通常，RAID 容量利用率在 50%～90%之间。

2. 高性能

RAID 的高性能受益于数据条带化技术。单个磁盘的 I/O 性能受到接口、带宽等计算机技术的限制，性能往往很有限，容易成为系统性能的瓶颈。通过数据条带化，RAID 将数据 I/O 分散到各个成员磁盘上，从而获得比单个磁盘成倍增长的聚合 I/O 性能。

3. 可靠性

可用性和可靠性是 RAID 的另一个重要特征。从理论上讲，由多个磁盘组成的 RAID 系统在可靠性方面应该比单个磁盘要差。这里有个隐含假定：单个磁盘故障将导致整个 RAID 不可用。RAID 采用镜像和数据校验等数据冗余技术，打破了这个假定。镜像是最为原始的冗余技术，把某组磁盘驱动器上的数据完全复制到另一组磁盘驱动器上，保证总有数据副本可用。比起镜像 50%的冗余开销，数据校验要小很多，它利用校验冗余信息对数据进行校验和纠错。RAID 冗余技术大幅提升了数据可用性和可靠性，保证了若干磁盘出错时不会导致数据的丢失，不影响系统的连续运行。

4. 可管理性

实际上，RAID 是一种虚拟化技术，它将多个物理磁盘驱动器虚拟成一个大容量的逻辑驱动器。对于外部主机系统来说，RAID 是一个单一的、快速可靠的大容量磁盘驱动器。这样，用户就可以在这个虚拟驱动器上来组织和存储应用系统数据。从用户应用角度看，可使存储系统简单易用，管理也很便利。由于 RAID 内部完成了大量的存储管理工作，管理员只需要管理单个虚拟驱动器，就可以节省大量的管理工作。RAID 可以动态增减磁盘驱动器，可自动进行数据校验和数据重建，这些都可以大大简化管理工作。

4.2　RAID 级别分类

RAID 每一个等级代表一种实现方法和技术，等级之间并无高低之分。在实际应用中，应当根据用户的数据应用特点，综合考虑可用性、性能和成本来选择合适的 RAID 等级，以

及具体的实现方式。

4.2.1 RAID0

RAID0 是一种简单的、无数据校验的数据条带化技术。但它和其他 RAID 级别有个很明显的区别——并不提供任何形式的冗余策略。RAID0 将所在磁盘条带化后组成大容量的存储空间，将数据分散存储在所有磁盘中，实现多块磁盘的同时读写，如图 4-1 所示。由于可以并发执行 I/O 操作，总线带宽得到充分利用，大幅提高了读写速度。因为不需要进行数据校验，通常情况下 RAID0 的性能在所有 RAID 等级中是最高的。理论上讲，在其他条件无限制的情况下，一个由 n 块磁盘组成的 RAID0，它的读写性能是单个磁盘性能的 n 倍。值得注意的是，如果在组成 RAID0 的磁盘中，有一块磁盘性能较差、容量较低，那么该 RAID0 的理论性能是由这块磁盘的性能、容量来作为计算基数的。

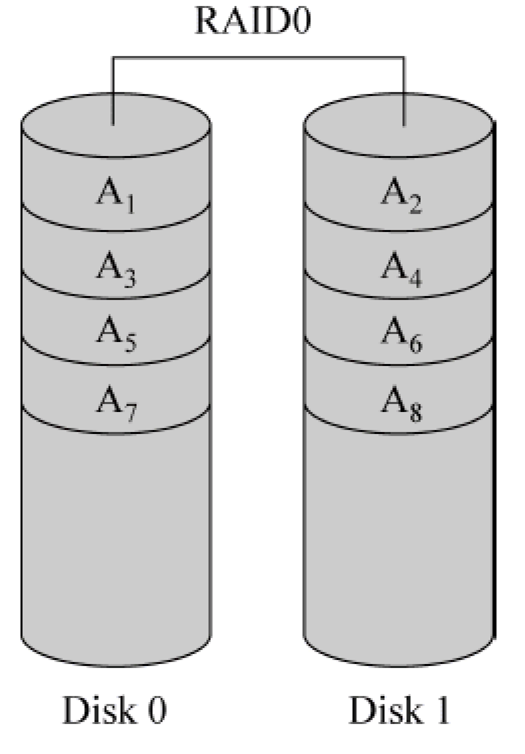


图 4-1 RAID0 示意图

RAID0 具有低成本、高读写性能、100%的高存储空间利用率等优点，但是它不提供数据冗余保护，一旦数据损坏，将无法恢复。因此，RAID0 一般适用于对性能要求严格但对数据安全性和可靠性不高的应用，如视频、音频存储、临时数据缓存空间等。

4.2.2 RAID1

RAID1 称为镜像，它将数据完全一致地分别写到工作磁盘和镜像磁盘上，它的磁盘空间利用率为 50%。因其实现原理的问题，RAID1 的读写性能较低，读写性能在 RAID 正常的情况下基本没有改变。RAID1 提供了最佳的数据保护，一旦工作磁盘发生故障，系统自动从镜像磁盘读取数据，不会影响用户的工作。工作原理如图 4-2 所示。

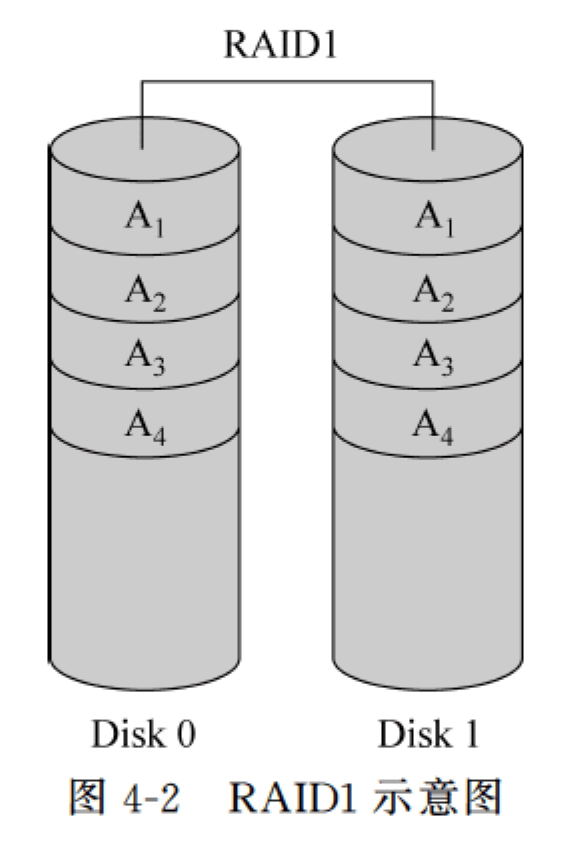


图 4-2 RAID1 示意图

RAID1 与 RAID0 刚好相反，是为了增强数据安全性使两块磁盘数据呈现完全镜像，从而达到安全性好、技术简单、管理方便。RAID1 拥有完全容错的能力，但实现成本高。比如，一个由 2 个 2TB 磁盘组成的 RAID1，其容量只有 2TB，但是 RAID1 能在 1 个磁盘损坏的情况下支持数据不丢失，其对数据安全的保护上是非常高的 RAID 级别。RAID1 应用于对顺序读写性能要求高以及对数据保护极为重视的应用，如对邮件系统、财务系统的数据保护。

4.2.3 RAID2

RAID2 称为纠错海明码磁盘阵列，其设计思想是利用海明码实现数据校验冗余。海明码是一种在原始数据中加入若干校验码来进行错误检测和纠正的编码技术，其中第 $2n$ 位（1，2，4，8，…）是校验码，其他位置是数据码。因此在 RAID2 中，数据按位存储，每块磁盘存储一位数据编码，磁盘数量取决于所设定的数据存储宽度，可由用户设定。如图 4-3 所示是

数据宽度为 4 的 RAID2，它需要 4 块数据磁盘和 3 块校验磁盘。如果是 64 位数据宽度，则需要 64 块数据磁盘和 7 块校验磁盘。可见，RAID2 的数据宽度越大，存储空间利用率越高，但同时需要的磁盘数量也越多。

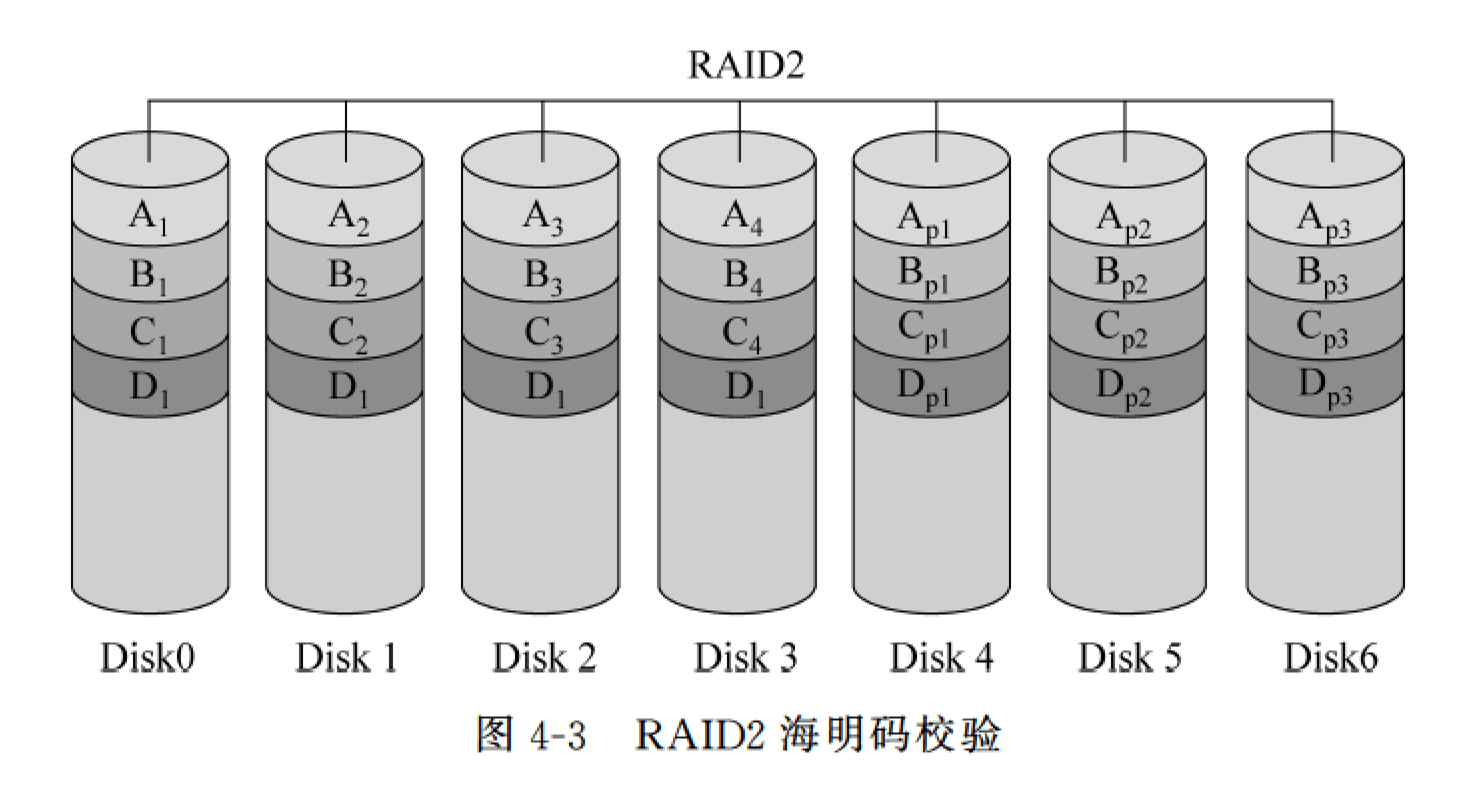


图 4-3　RAID2 海明码校验

海明码自身具备纠错能力，因此 RAID2 可以在数据发生错误的情况下纠正错误，保证数据的安全性。它的数据传输性能相当高，设计复杂性要低于后面介绍的 RAID3、RAID4 和 RAID5。

但是，海明码的数据冗余开销太大，而且 RAID2 的数据输出性能受阵列中最慢磁盘驱动器的限制。再者，海明码是按位运算，RAID2 数据重建非常耗时。由于这些显著的缺陷，再加上大部分磁盘驱动器本身都具备了纠错功能，因此 RAID2 在实际中很少应用，没有形成商业产品，目前主流存储磁盘阵列均不提供 RAID2 支持。

4.2.4　RAID3

RAID3 是使用专用校验盘的并行访问阵列，它采用一个专用的磁盘作为校验盘，其余磁盘作为数据盘，数据按位可字节的方式交叉存储到各个数据盘中，如图 4-4 所示。RAID3 至少需要 3 个磁盘，不同磁盘上同一带区的数据作 XOR 校验，校验值写入校验盘中。RAID3 正常时读性能与少 1 个磁盘的 RAID0 完全一致，并行从多个磁盘条带读取数据，性能较高，同时还提供了数据容错能力，在 1 个硬盘出现故障的情况下支持数据不丢失。向

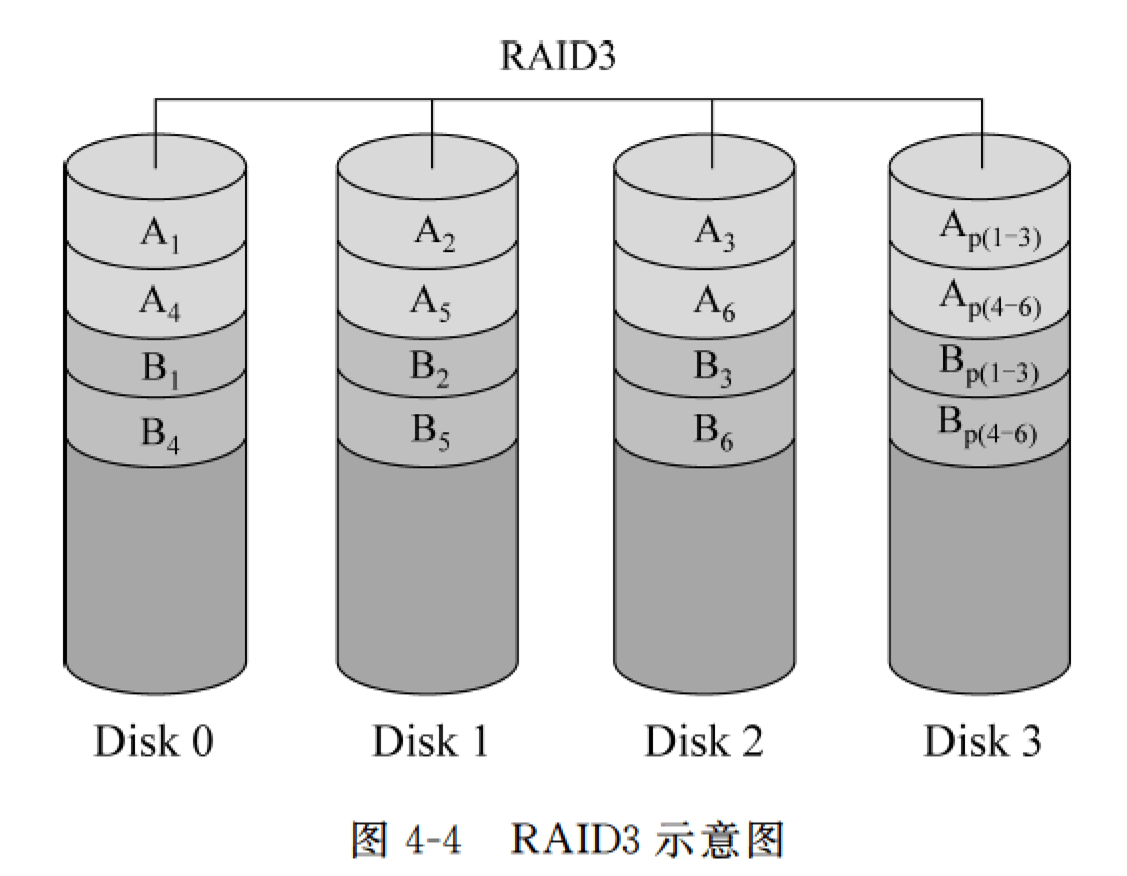


图 4-4　RAID3 示意图

RAID3 写入数据时，因增加了计算和写入校验值的过程，所以写性能较低。

如果 RAID3 中某一磁盘出现故障，不会影响数据读取，可以借助校验数据和其他完好数据来重建数据。假如所要读取的数据块正好位于失效磁盘上，则系统需要读取所有同一条带的数据块，并根据校验值重建丢失的数据，系统性能将受到影响。当故障磁盘被更换后，RAID 系统按相同的方式重建故障盘中的数据至新磁盘上。

RAID3 只有 1 个校验盘，阵列的存储空间利用率较高，再加上并行访问的特征，能够为高带宽的大量读写提供高性能，适用大容量数据的顺序访问应用，如影像处理、流媒体服务等。但是，随着 RAID5 算法的不断改进，在大数据量读取时能够模拟 RAID3，而且 RAID3 在出现坏盘时会出现性能大幅下降的情况，因此现在的 RAID3 大部分已被 RAID5 替代了。

4.2.5 RAID4

RAID4 与 RAID3 的原理大致相同，区别在于条带化的方式不同。RAID4（图 4-5）按照块的方式来组织数据，写操作只涉及当前数据盘和校验盘两个盘，多个 I/O 请求可以同时得到处理，提高了系统性能。RAID4 按块存储可以保证单块的完整性，可以避免受到其他磁盘上同条带产生的不利影响。

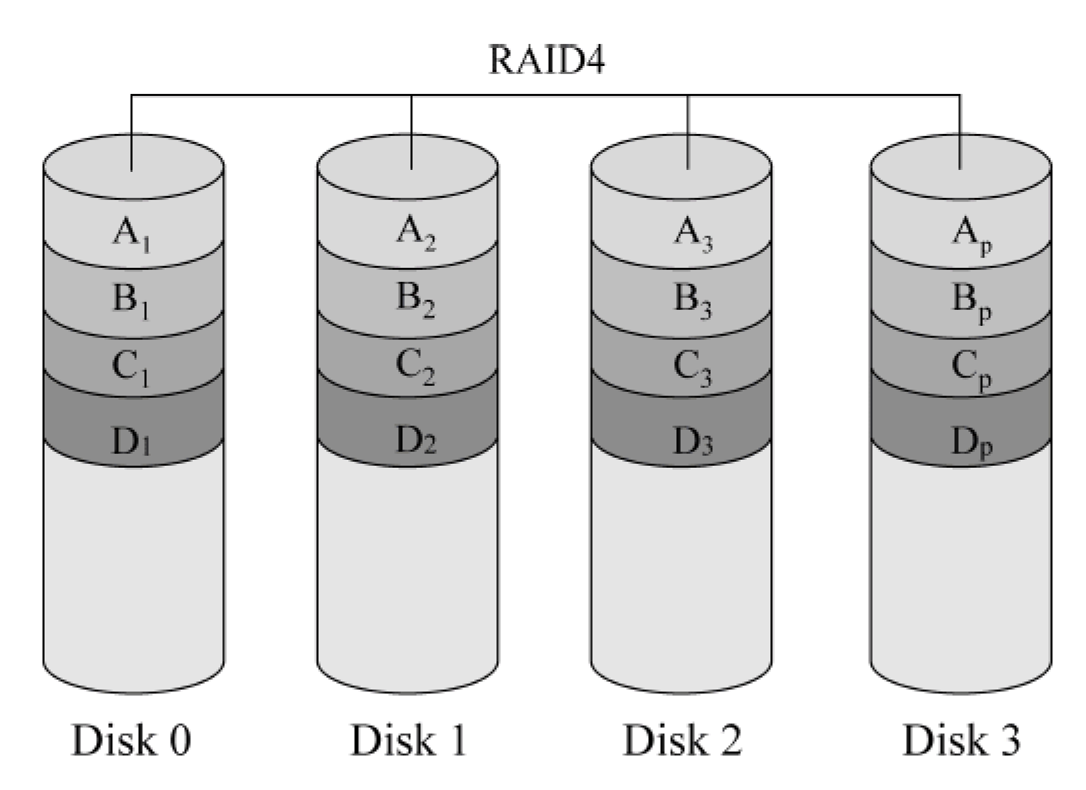


图 4-5　带有专用块级校验的数据条带

RAID4 在不同磁盘上的同级数据块同样使用 XOR 校验，结果存储在校验盘中。写入数据时，RAID4 按这种方式把各磁盘上的同级数据的校验值写入校验盘，读取时进行即时校验。因此，当某个磁盘的数据块损坏，RAID4 可以通过校验值以及其他磁盘上的同级数据块进行数据重建。

RAID4 提供了非常好的读性能，但单一的校验盘往往成为系统性能的瓶颈。对于写操作，RAID4 只能一个磁盘一个磁盘地写，并且还要写入校验数据，因此写性能比较差。而且随着成员磁盘数量的增加，校验盘的系统瓶颈将更加突出。正是如上这些限制和不足，RAID4 在实际应用中很少见，主流存储产品也很少使用 RAID4 保护。

4.2.6 RAID5

RAID5 应该是目前最常见的 RAID 等级，它的原理与 RAID3 相似，区别在于校验数据分布在阵列中的所有磁盘上，而没有采用专门的校验磁盘，如图 4-6 所示。对于数据和校验

数据，它们的写操作可以同时发生在完全不同的磁盘上。因此，RAID5 不存在 RAID3 中的并发写操作时的校验盘性能瓶颈问题。另外，RAID5 还具备很好的扩展性。当阵列磁盘数量增加时，并行操作量的能力也随之增长，从而拥有更高的容量以及更高的性能。

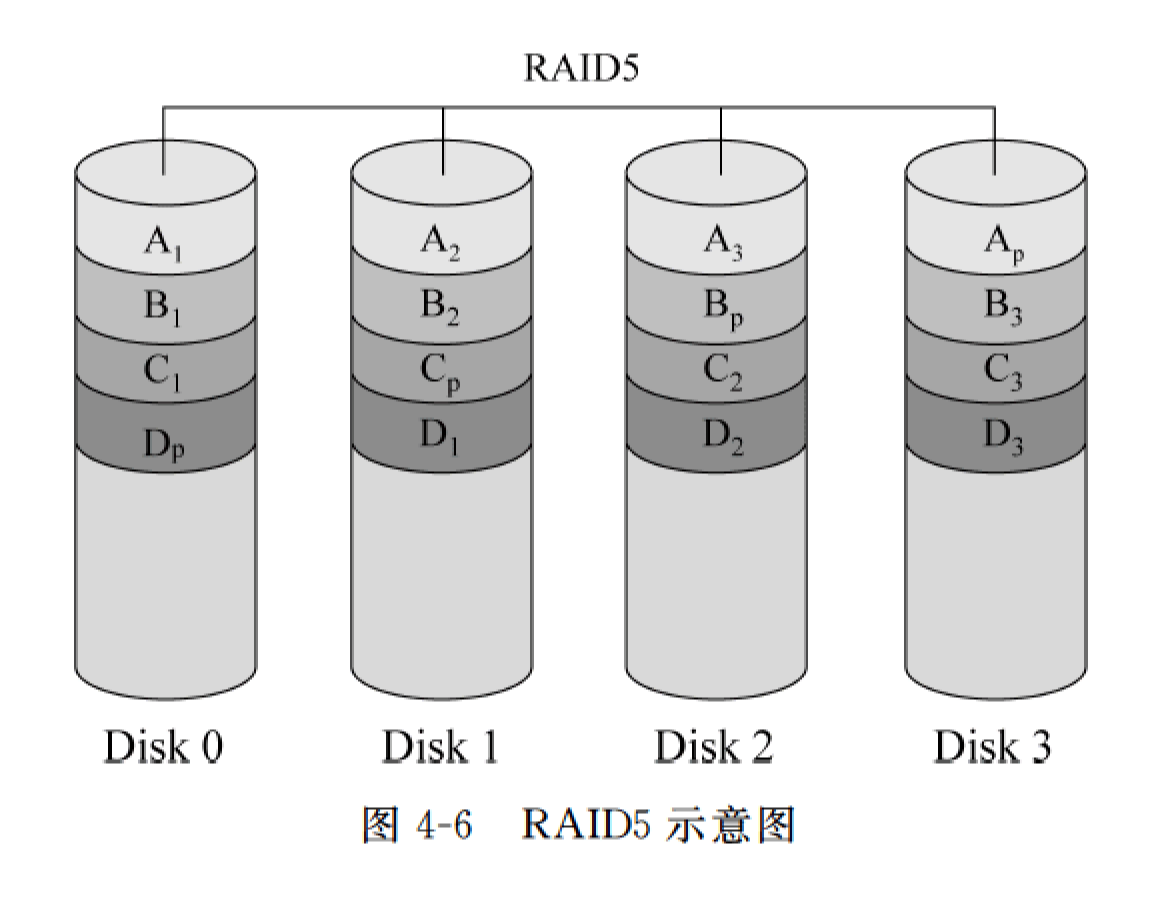


图 4-6　RAID5 示意图

RAID5 的磁盘上同时存储数据和校验数据，数据块和对应的校验信息保存在不同的磁盘上。当一个数据盘损坏时，系统可以根据同一条带的其他数据块和对应的校验数据来重建损坏的数据。与其他 RAID 等级一样，重建数据时，RAID5 的性能会受到较大的影响。

RAID5 兼顾存储性能、数据安全和存储成本等各方面因素，它可以理解为 RAID0 和 RAID1 的折中方案，是目前综合性能最佳的数据保护解决方案。RAID5 基本上可以满足大部分的存储应用需求，数据中心大多采用它作为应用数据的保护方案。

4.2.7　RAID6

前面所述的各个 RAID 等级都只能保让因单个磁盘失效而造成的数据丢失。如果两个磁盘同时发生故障，数据将无法恢复。RAID6 引入双重校验的概念，它可以保证阵列中同时出现两个磁盘失效时阵列仍能够继续工作，不会发生数据丢失。RAID6 等级是在 RAID5 的基础上为了进一步增强数据保护而设计的一种 RAID 方式，它可以看作是一种扩展的 RAID5 等级。

RAID6 不仅要支持数据的恢复，还要支持校验数据的恢复，因此实现代价很高，控制器的设计也比其他等级更复杂、更昂贵。RAID6 思想最常见的实现方式是采用两个独立的校验算法，假设称为 P 和 Q，校验数据可以分别存储在两个不同的校验盘上，或者分散存储在所有成员磁盘中。当两个磁盘同时失效时，即可通过求解两元方程来重建两个磁盘上的数据。如图 4-7 所示。

RAID6 具有快速的读取性能、更高的容错能力。但是，它的成本要高于 RAID5，写性能也比 RAID5 差一些。因此，RAID6 主要用于对数据安全等级要求较高的场合。它一般是替代 RAID10 方案的经济性选择。

4.2.8　RAID 组合等级

标准 RAID 等级各有优势和不足。自然地，我们会想到把多个 RAID 等级组合起来，实

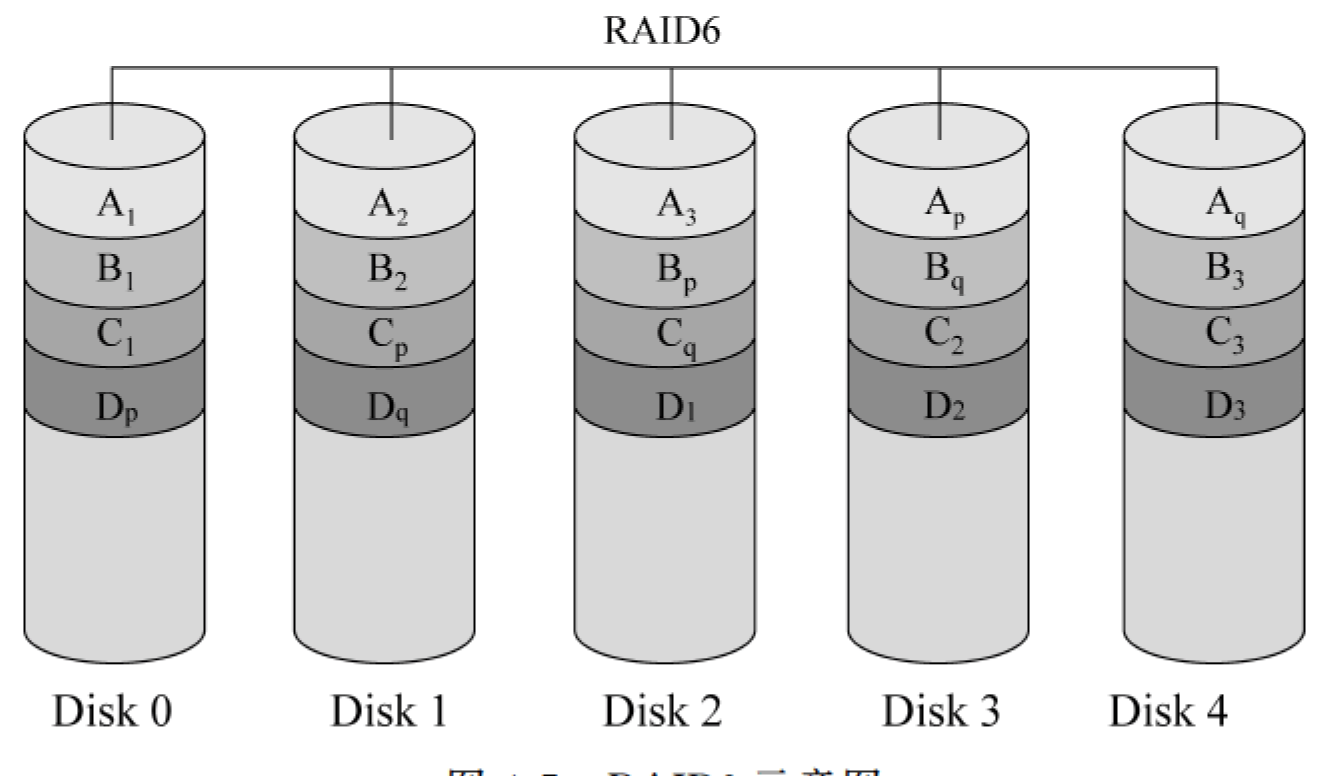


图 4-7　RAID6 示意图

现优势互补，弥补相互的不足，从而达到在性能、数据安全性等指标上更高的 RAID 系统。目前在业界和学术研究中提到的 RAID 组合等级主要有 RAID00、RAID01、RAID10、RAID100、RAID30、RAID50、RAID53、RAID60，但实际得到较为广泛应用的只有 RAID01 和 RAID10 两个等级。当然，组合等级的实现成本一般都非常昂贵，只是在少数特定场合应用。

1．RAID00

简单地说，RAID00 是由多个成员 RAID0 组成的高级 RAID0。它与 RAID0 的区别在于，RAID0 阵列替换了原先的成员磁盘。可以把 RAID00 理解为两层条带化结构的磁盘阵列，即对条带再进行条带化。这种阵列可以提供更大的存储容量、更高的 I/O 性能和更好的 I/O 负均衡。

2．RAID01 和 RAID10

一些文献把这两种 RAID 等级看作是等同的，本书认为是不同的。RAID01 是先做条带化再作镜像，本质是对物理磁盘实现镜像；而 RAID10 是先做镜像再作条带化，是对虚拟磁盘实现镜像。相同的配置下，通常 RAID01 比 RAID10 具有更好的容错能力，原理如图 4-8 所示。

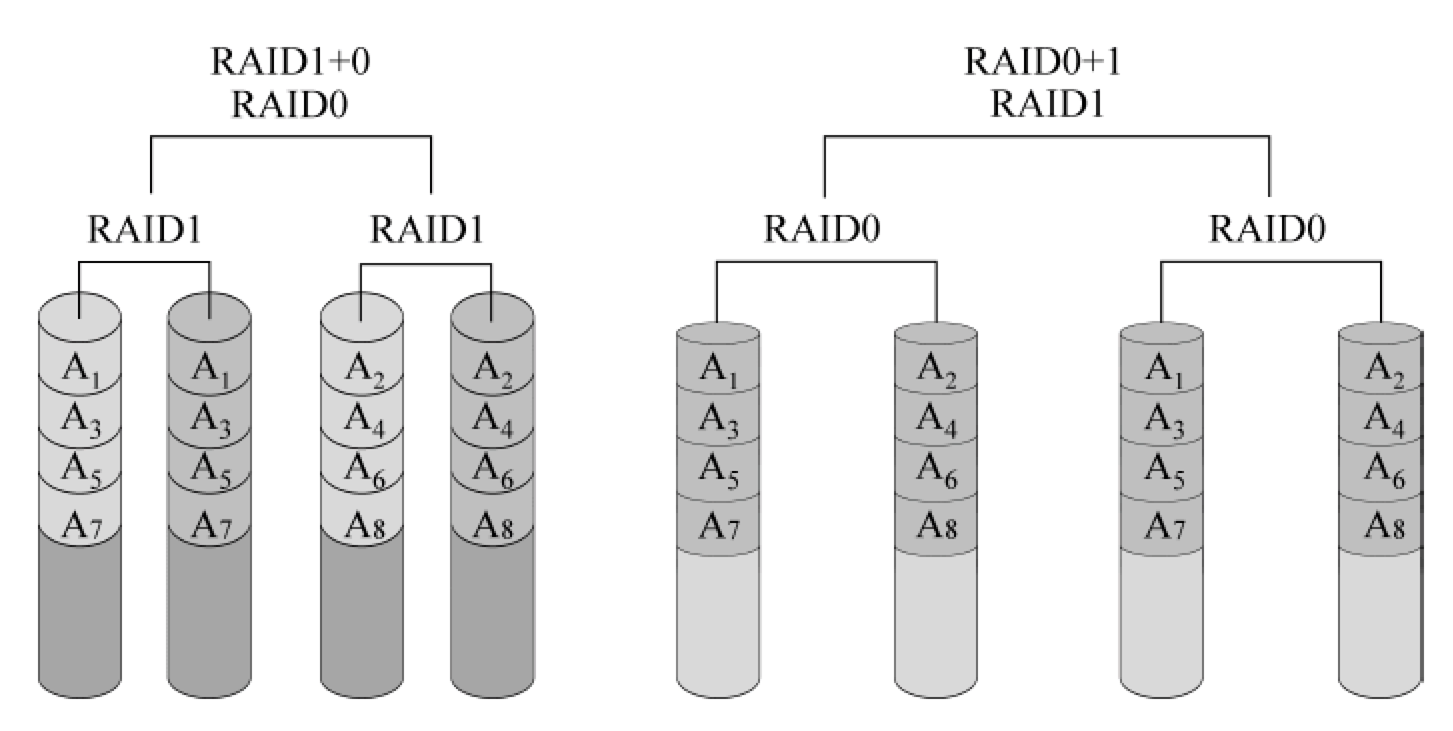


图 4-8　典型的 RAID01(左)和 RAID10(右)模型

RAID01 兼备了 RAID0 和 RAID1 的优点，它先用两块磁盘建立镜像，然后再在镜像内部做条带化。RAID01 的数据将同时写入两个磁盘阵列中，如果其中一个阵列损坏，仍可继

续工作,保证数据安全性的同时又提高了性能。RAID01 和 RAID10 内部都含有 RAID1 模式,因此整体磁盘利用率均仅为 50%。

3. RAID100

RAID100 通常看作 RAID 1+0+0,有时也称为 RAID 10+0,即条带化的 RAID10,原理如图 4-9 所示。RAID100 的缺陷与 RAID10 相同,任意一个 RAID1 损坏一个磁盘不会发生数据丢失,但是剩下的磁盘存在单点故障的危险。最顶层的 RAID0,即条带化任务,通常由软件层来完成。

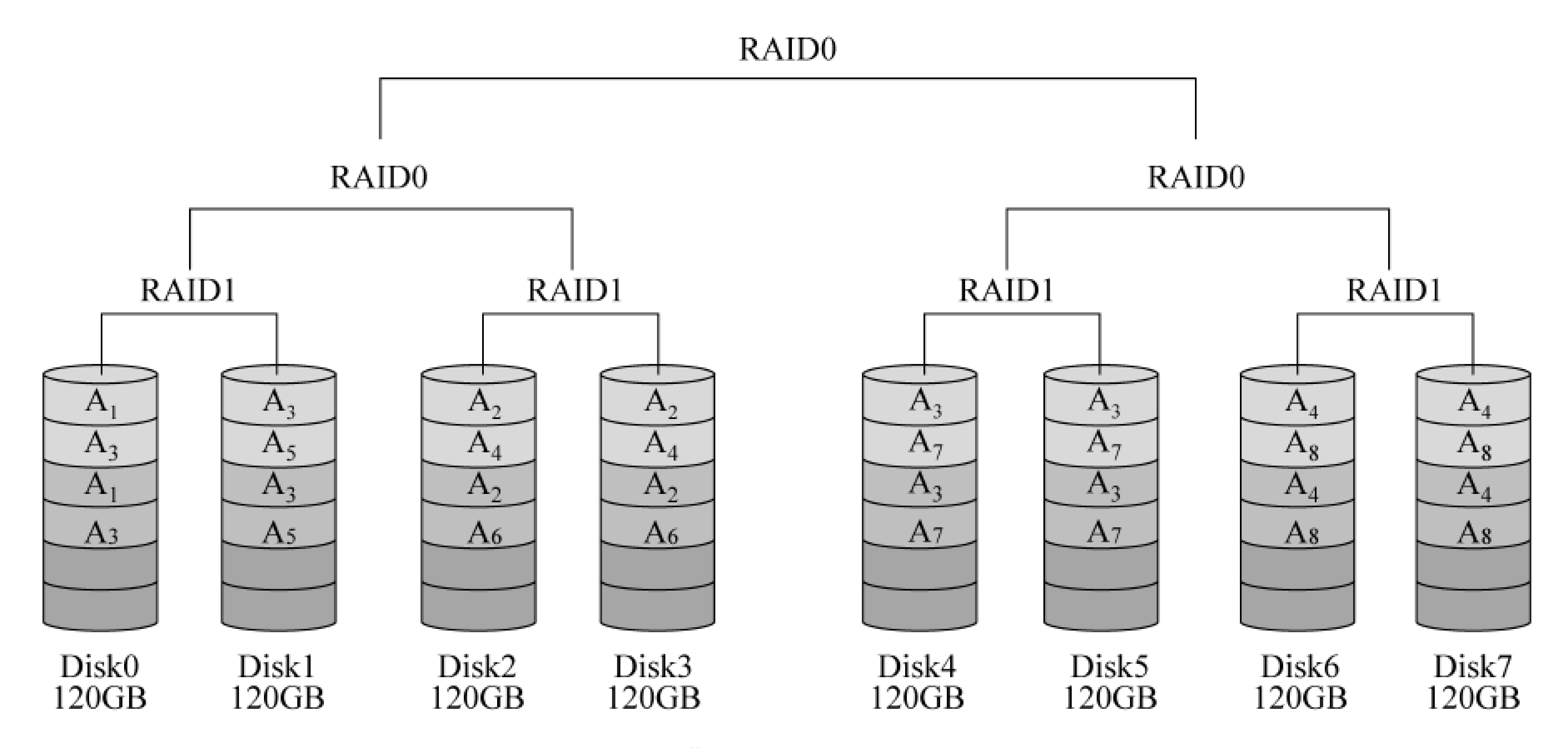


图 4-9　典型的 RAID100 模型

RAID100 突破了单个 RAID 控制器对物理磁盘数量的限制,可以获得更高的 I/O 负载均衡,I/O 压力分散到更多的磁盘上,进一步提高随机读性能,并有效降低热点盘故障风险。因此,RAID100 通常是大数据库的最佳选择。

4. RAID30(RAID53)、RAID50 和 RAID60

这三种 RAID 等级与 RAID00 原理基本相同,区别在于成员“磁盘”换成了 RAID3、RAID5 和 RAID6,分别如图 4-10～图 4-12 所示。其中,RAID30 通常又称为 RAID53。其

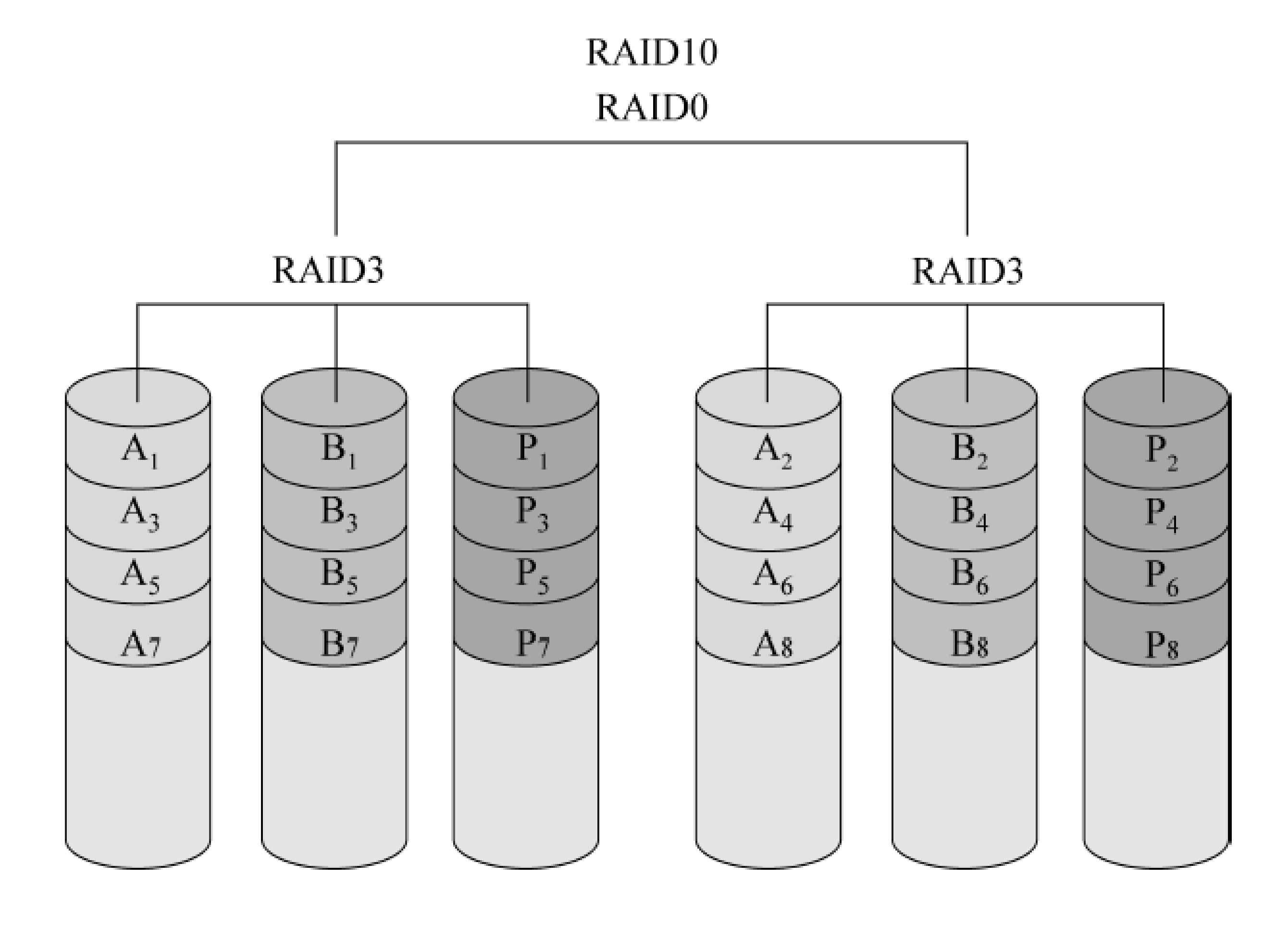


图 4-10　典型的 RAID30 模型

实，可把这些等级 RAID 统称为 RAID X0 等级，X 可为标准 RAID 等级，甚至组合等级（如 RAID100）。利用多层 RAID 配置，充分利用 RAID X 与 RAID0 的优点，从而获得在存储容量、数据安全性和 I/O 负载均衡等方面性能的大幅提升。

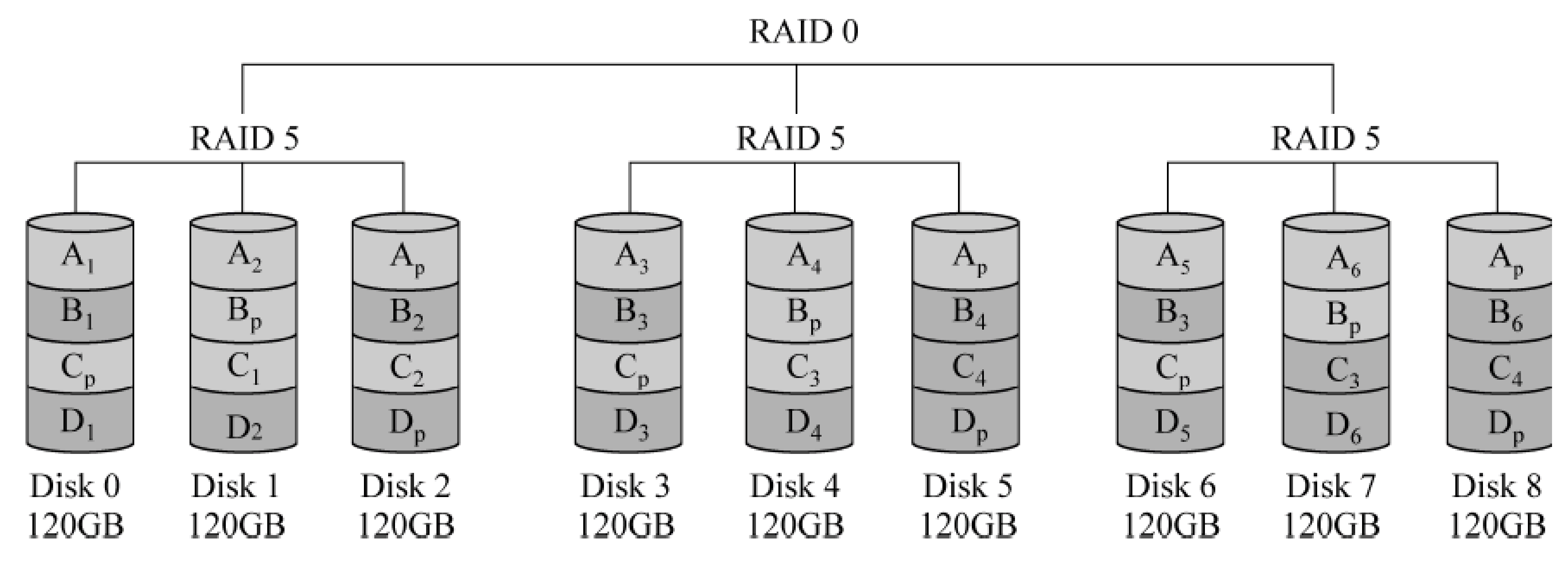


图 4-11　典型的 RAID50 模型

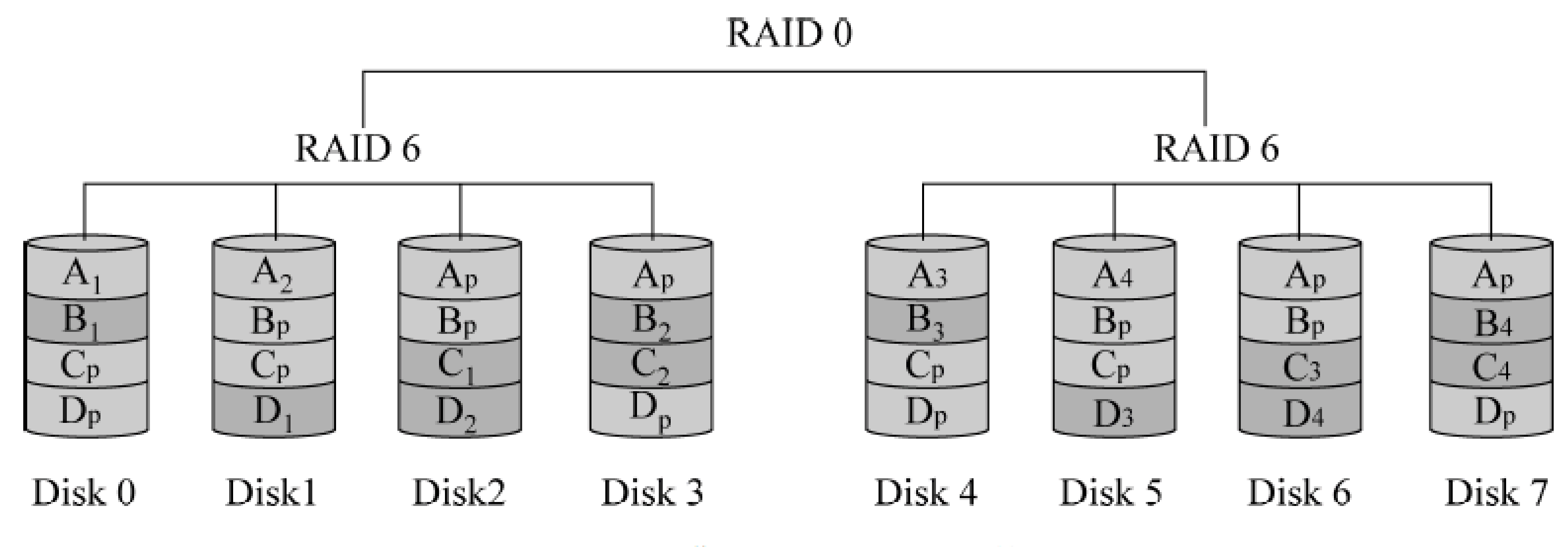


图 4-12　典型的 RAID60 模型

4.2.9　非标准 RAID 等级

虽然标准 RAID 和组合 RAID 在具体实现上存在一定程度的不同，但与标准规范是保持一致或兼容的。然而除此之外，一些存储厂商还实现了非标准的 RAID 等级，往往都是公司私有的产品。这里简单介绍几个非标准 RAID 等级。

1. RAID7

RAID7 全称叫 Optimized Asynchrony for High I/O Rates as well as high Data Transfer Rates（最优化的异步高 I/O 速率和高数据传输率），它与以前我们见到 RAID 级别具有明显的区别。RAID 7 完全可以理解为一个独立存储计算机，它自身带有操作系统和管理工具，完全可以独立运行。

RAID7 等级是至今为止理论上性能最高的 RAID 模式，因为它从组建方式上就已经和以往的方式有了重大的不同。你会发现在，以往一个硬盘是一个组成阵列的“柱子”，而在 RAID7 中，多个硬盘组成一个“柱子”，它们都有各自的通道，也正因为如此，你可以把这个图分解成一个个硬盘连接在主通道上，只是比以前的等级划分更细致了。这样做的好处就是在读/写某一区域的数据时，可以迅速定位，而不会因为以往因单个硬盘的限制同一时间只能访问该数据区的一部分，在 RAID7 中，以前的单个硬盘相当于分割成多个独立的硬盘，有自己的读写通道，效率也就不言自明了。

RAID7 不仅仅是一种技术，还是一种存储计算机（Storage Computer）。RAID7 存储计算机操作系统（Storage Computer Operating System）是一套实时事件驱动操作系统，主要用来进行系统初始化和安排 RAID7 磁盘阵列的所有数据传输，并把它们转换到相应的物理存储驱动器上。通过自身系统中的阵列电脑板来设定和控制读写速度，存储计算机操作系统可使主机 I/O 传递性能达到最佳。如果一个磁盘出现故障，还可自动执行恢复操作，并可管理备份磁盘的重建过程，如图 4-13 所示。

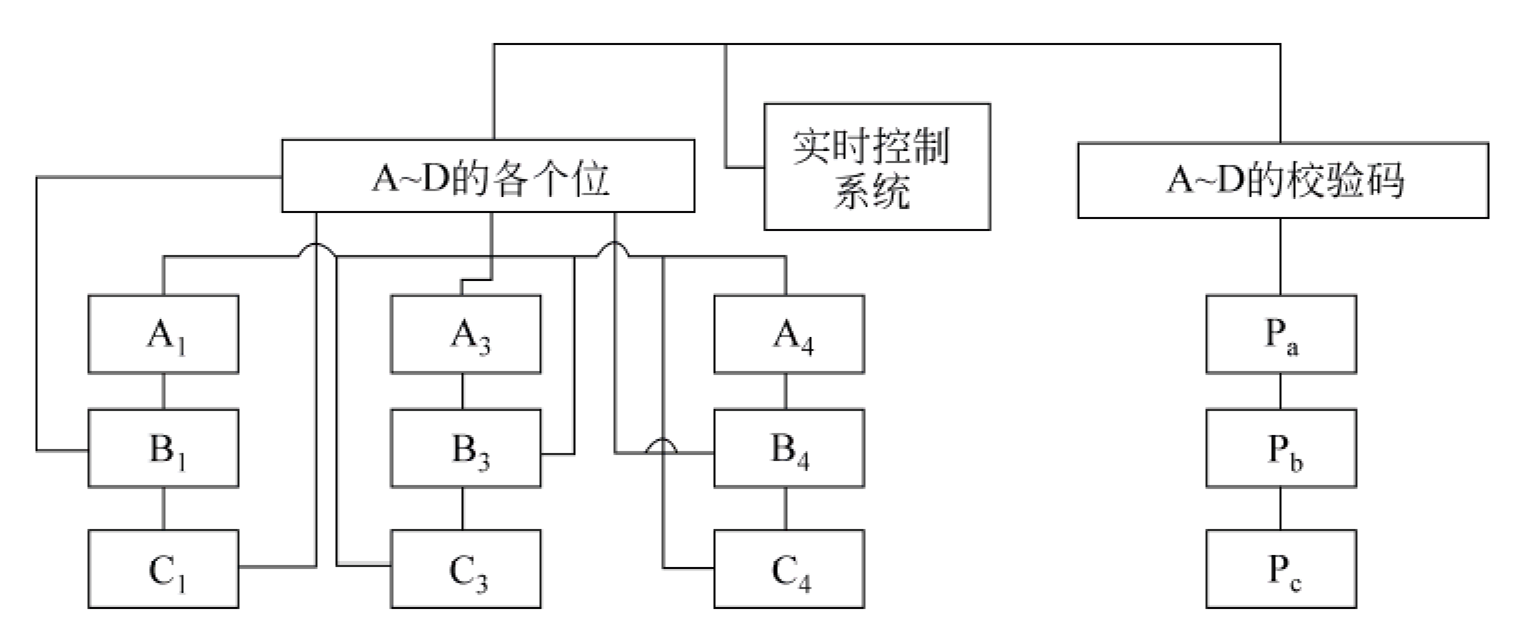


图 4-13　RAID7 结构示意图

RAID7 突破了以往 RAID 标准的技术架构，采用了非同步访问，极大地减轻了写数据的瓶颈，提高了 I/O 速度。所谓非同步访问，即 RAID7 的每个 I/O 接口都有一条专用的高速通道，作为数据或控制信息的流通路径，因此可独立地控制自身系统中每个磁盘的数据存取。如果 RAID7 有 N 个磁盘，那么除去一个校验盘（用作冗余计算）外，可同时处理 $N-1$ 个主机系统随机发出的读/写指令，从而显著地改善了 I/O 应用。RAID7 系统内置实时操作系统还可自动对主机发送过来的读/写指令进行优化处理，以智能化方式将可能被读取的数据预先读入快速缓存中，从而大大减少了磁头的转动次数，提高了 I/O 速度。RAID7 可帮助用户有效地管理日益庞大的数据存储系统，并使系统的运行效率提高至少一倍以上，满足了各类用户的不同需求。

下面介绍 RAID7 的特点。

RAID7 可完全独立于主机运行，不占用主机 CPU 资源。

RAID7 是一套实时事件驱动操作系统，主要用来进行系统初始化和安排 RAID7 磁盘阵列的所有数据传输，并把它们转换到相应的物理存储驱动器上。通过存储计算机操作系统来设定和控制读写速度，可使主机 I/O 传递性能达到最佳。如果一个磁盘出现故障，还可自动执行恢复操作，并可管理备份磁盘的重建过程。

RAID7 已经被 Storage Computer Corporation 注册了商标，下面就让我们作一下总结。

（1）优点

① 全面写入的性能领先其他硬盘性能 25%～90%并且强于其他阵列 1.5～6 倍。

② 主机接口通过可升级的连通性来增加传输带宽。

③ 在小规模用户读取操作时，Cache 的命中率极高，几乎可以将寻址时间变相降低为零。

④ 阵列中的磁盘数量越多，写入效率提高越大，读取时寻址时间越短。

⑤ 没有额外的带宽用于效验操作。

(2) 缺点

① 很可能造成一个卖主一个方案的局面。

② 存储容量中,每 MB 成本极高。

③ 相对而言,非常短的保修期。

④ 大多数用户可能都用不到。

⑤ 必须要有 UPS 的配合以保证意外断电时 Cache 中的数据顺利保存。

然而,RAID7 的设计与相应的组成规模注定了它是一揽子承包计划。总体上说,RAID7 是一个整体的系统,RAID7 的设计与相应的组成规模注定了它是一揽子承包计划。总体上说,RAID7 是一个整体的系统,有自己的操作系统,有自己的处理器,有自己的总线,而不是通过简单的插卡就可以实现的。归纳起来,RAID7 的主要特性如下。

(1) 所有的 I/O 传输都是异步的,因为它有自己独立的控制器和带有 Cache 的接口,与系统时钟并不同步。

(2) 所有的读与写的操作都将通过一个带有中心 Cache 的高速系统总线,我们称之为 X-Bus。

(3) 专用的校验硬盘可以用于任何通道。

带有完整功能的即时操作系统内嵌于阵列控制微处理器中,这是 RAID7 的心脏,它负责各通道的通信以及 Cache 的管理,这也是它与其他等级最大的不同之处。

2. RAID-DP

按照 SNIA 最新的 RAID6 定义,双重数据校验的磁盘阵列都可归为 RAID6 等级。NetApp 公司按照 RAID6 的定义实现了 RAID-DP,使用双重的数据校验来保护数据,可以保证两块磁盘同时损坏的情况下不发生数据丢失。与该公司的 RAID4 实现对比,传统的 RAID6 实现会致使系统性能损失 30%左右,而 RAID-DP 的性能下降低于 2%。上层文件系统的请求首先写入后端的 NVRAM 中,确保即使在掉电的情况下也不会有任何数据丢失。因此,数据块不会立即更新。当执行新来的写操作,会对写操作进行聚集,然后存储控制器尝试一次性写入包括校验数据在内的整个数据条带。RAID-DP 提供了比 RAID10 更好的数据保护,性能却不低于 RAID10。对于相同大小的 RAID 组,在大多数情况下,RAID-DP 没有受到传统 RAID6 即时更新数据块的挑战,并提供更多的磁盘进行读写。它甚至允许磁盘固件实时更新而不发生任何中断。

3. RAID1.5

这是 HighPoint 公司的 RAID 产品,有时也被错误地称为 RAID1.5。RAID1.5 仅使用两个磁盘驱动器同时进行数据条带化和镜像,数据可以同时从两块磁盘进行读取。这其中的大部分工作都由硬件来完成,而非驱动程序。Linux、Solaris 等操作系统实现的 RAID1 也可以实现同时从两块磁盘进行读取数据,因此 RAID1.5 并不优于传统的 RAID1。

4. RAID5E、RAID5EE 和 RAID6E

这种概念首次在 IBMServer RAID 中被提出,E 是 Enhanced 的首字母。它们分别是对 RAID5 和 RAID6 的增强,增加了热冗余磁盘驱动器,冗余磁盘与其他磁盘一起进行数据块编排。这种设计使得 I/O 可以分散到包括热冗余在内的所在磁盘,从而减小单块磁盘的 I/O 带宽,提供更高的性能。然而,热冗余磁盘不能够被多个阵列共享。

在实现中,实际上不存在专用的热冗余磁盘,就像 RAID5 和 RAID6 中没有专用的校验

磁盘一样，所有的冗余数据块分布在成员磁盘中。例如，一个有10块磁盘的RAID5E，包括80%数据块、10%的冗余数据块和10%的校验数据。对于RAID5E和RAID6E，冗余数据块位于阵列尾部，而RAID5EE则分布在整个RAID中。如果RAID5E/5EE中发生一块磁盘损坏，则系统会自动降级并重建至标准的RAID5。这一过程中，I/O操作非常密集，并且需要花费大量时间，为几个小时甚至几天，根据阵列的具体配置而异。当损坏磁盘被替换后，系统则又会自动升级并重建至原先的RAID5E/5EE，同时非常耗时。在上面的重建过程中，数据没有冗余保护。由于系统升级和降级时I/O活动密集且所需时间过长，因此实际应用中成员磁盘数据限制在4～8块。一旦超过8块磁盘，由于损坏磁盘的重建耗时和重建中发生第二块磁盘损坏造成的数据丢失，RAID5E/5EE所获得的性能提升和其他获益都将严重降低。

5. RAID S(Parity RAID)

RAID S是EMC公司的Symmetrix存储系统所使用的条带化校验RAID。该系统中，每个卷位于单独的物理磁盘上，多个卷组合进行数据校验。EMC最早引入了RAID S概念，后来改名为Parity RAID并应用于Symmetrix DMX平台。EMC现在也为Symmetrix DMX提供标准的RAID5，RAID S已经不再EMC产品中使用。

6. Intel Matrix RAID

Matrix RAID是Intel ICH6R和后继的南桥芯片的一个重要特征，可以通过RAID BIOS进行访问。它使用两块磁盘或者控制器能支持的最多磁盘，它的显著特征是允许RAID0、RAID1、RAID5、RAID10多种数据卷混合共存，每块磁盘的指定部分分配给相应的RAID卷。Matrix RAID主要用于改善性能和数据完整性，实际应用中可以将操作系统应用于小的RAID0，而大的RAID1存储关键数据以及用户数据。海量的流媒体数据容易发生数据丢失，可以考虑使用这种RAID。Linux的MD RAID也可以实现类似的功能。

7. Linux MD RAID 10

RAID 10是Linux内核所支持的软RAID等级之一，它还支持RAID0～RAID6等级别。软RAID驱动程序通常通过构造典型的RAID1+0阵开来实现RAID10，2.6.9以后的内核也可作为单独的级别来实现。

MD RAID10支持重复数据块的近布局和远布局两种模式。近布局与标准RAID10相同，镜像数据块相邻存储。对于n重镜像的k路条带，不要求k为n的整倍数。两重镜像的2、3、4路条带的MD RAID10分布相当于RAID1、RAID-1E和RAID10。远布局模式下，所有磁盘被划分为f(f=镜像数)个数据存储区，重复数据块相对于原始数据块具有一个磁盘和若干依偏移的距离，即保存在下一个磁盘对应存储区的偏移位置。这种设计能够提高镜像阵列的条带性能，有效提高顺序和随机读性能，但对写性能没有显著提升。许多应该通常具有读密集而写稀疏的特点，RAID10适合此类数据应用。需要指出的是，近布局和远布局两种模式可以同时使用，这种情况下将有$n \times f$个数据副本。

8. IBM Server RAID 1E

IBM公司的Server RAID阵列卡系列支持任意数量驱动器上的两路镜像，多个磁盘对数据块进行轮转镜像。这种配置能够对不相邻磁盘驱动器发生的损坏进行容错，其他的存储系统也支持这种模式，比如SUN公司的StorEdge T3。

9. RAID-K

Kaleidescape 公司实现了一种称为 RAID-K[16]的 RAID 类型。RAID-K 与 RAID4 相似，但不对文件数据进行块级的条带化处理，它企图将整个电影或音乐集合完整地存储在单个磁盘上。另外，它的冗余校验信息可存储在多个磁盘上，从而适应由多个容量不同的磁盘所组成的逻辑磁盘。而且，冗余数据包含比校验信息更多的数据，用于获取更高的容错性。这些特征可以为影像、音乐提供更好的性能，增加数据存储的安全性。RAID-K 还可以允许用户以增量方式扩充存储容量，能够增加容量更大的磁盘，甚至它还可以增加包含数据（仅限影像和音乐）的磁盘。RAID-K 会自动把这些磁盘组建成 RAID-K 阵列和 Kaleidescape 文件系统。

10. RAID-Z

RAID-Z 是集成在 SUN 公司 ZFS 文件系统中的一种与 RAID5 相似的 RAID 模式。利用写时复制策略，RAID-Z 避免了 RAID5 的写操作困境（即更新数据同时需要更新校验数据），它不用新数据覆盖旧数据，而是把新数据写到新位置并自动更新数据指针。对于小的写操作，仅仅执行完全的写条带操作，有效避免“读取－更改－写回”的操作需求。另外，还可以直接对小写操作使用镜像替换校验进行保护，因为文件系统了解下层存储结构，可以在必要时分配额外存储空间。ZFS 还实现了 RAID-Z2，提供类似于 RAID6 的双重校验保护能力，可以保证磁盘发生损坏时不发生数据丢失。后来 ZFS 加入了三重校验 RAID 支持，或许称为 RAID-Z3。

4.2.10 RAID 的实现方式

通常计算机功能既可以由硬件来实现，也可以由软件来实现。对于 RAID 系统而言，自然也不例外，它可以采用软件方式实现，也可以采用硬件方式实现，或者采用软硬结合的方式实现。

1. 软 RAID

软 RAID 没有专用的控制芯片和 I/O 芯片，完全由操作系统和 CPU 来实现 RAID 的功能。现代操作系统基本上都提供对软 RAID 的支持，通过在磁盘设备驱动程序上添加一个软件层，提供一个物理驱动器与逻辑驱动器之间的抽象层。目前，操作系统支持的最常见的 RAID 等级有 RAID0、RAID1、RAID10、RAID01 和 RAID5 等。比如，Windows Server 支持 RAID0、RAID1 和 RAID5 三种等级，Linux 支持 RAID0、RAID1、RAID4、RAID5、RAID6 等，Mac OS X Server、FreeBSD、NetBSD、OpenBSD、Solaris 等操作系统也都支持相应的 RAID 等级。

软 RAID 的配置管理和数据恢复都比较简单，但是 RAID 所有任务的处理完全由 CPU 来完成，如计算校验值，所以执行效率比较低下，这种方式需要消耗大量的运算资源，支持的 RAID 模式较少，很难广泛应用。

软 RAID 由操作系统来实现，因此系统所在分区不能作为 RAID 的逻辑成员磁盘，软 RAID 不能保护系统盘 D。对于部分操作系统而言，RAID 的配置信息保存在系统信息中，而不是单独以文件形式保存在磁盘上。这样当系统意外崩溃而需要重新安装时，RAID 信息就会丢失。另外，磁盘的容错技术并不等于完全支持在线更换、热插拔或热交换，能否支持错误磁盘的热交换与操作系统的实现相关，有的操作系统热交换。

2. 硬 RAID

硬 RAID 拥有自己的 RAID 控制处理与 I/O 处理芯片，甚至还有阵列缓冲，对 CPU 的占用率和整体性能是三类实现中最优的，但实现成本也最高的。硬 RAID 通常都支持热交换技术，在系统运行下更换故障磁盘。

硬 RAID 包含 RAID 卡和主板上集成的 RAID 芯片，服务器平台多采用 RAID 卡。RAID 卡由 RAID 核心处理芯片（RAID 卡上的 CPU）、端口、缓存和电池 4 部分组成。其中，端口是指 RAID 卡支持的磁盘接口类型，如 IDE/ATA、SCSI、SATA、SAS、FC 等接口。

3. 软硬混合 RAID

软 RAID 性能欠佳，而且不能保护系统分区，因此很难应用于桌面系统。而硬 RAID 成本非常昂贵，不同 RAID 相互独立，不具有互操作性。因此，人们采取软件与硬件结合的方式来实现 RAID，从而获得在性能和成本上的一个折中，即较高的性价比。

这种 RAID 虽然采用了处理控制芯片，但是为了节省成本，芯片往往比较廉价且处理能力较弱，RAID 的任务处理大部分还是通过固件驱动程序的 CPU 来完成。

4.3　RAID 的应用选择

RAID 等级的选择主要有三个因素，即数据可用性、I/O 性能和成本。目前，在实际应用中常见的主流 RAID 等级是 RAID0，RAID1，RAID3，RAID5，RAID6，RAID01 和 RAID10，它们之间的技术对比情况如表 4-1 所示。如果不要求可用性，可选择 RAID0 以获得高性能。如果可用性和性能是重要的，而成本不是一个主要因素，则根据磁盘数量选择 RAID1。如果可用性成本和性能都同样重要，则根据一般的数据传输和磁盘数量选择 RAID3 或 RAID5。在实际应用中，应当根据用户的数据应用特点和具体情况，综合考虑可用性、性能和成本来选择合适的 RAID 等级。

表 4-1　主流 RAID 等级技术对比

RAID 等级	RAID0	RAID1	RAID3	RAID5	RAID6	RAID10
别名	条带	镜像	专用奇偶校验条带	分布奇偶校验条带	双重奇偶校验条带	镜像加条带
容错性	无	有	有	有	有	有
冗余类型	无	有	有	有	有	有
热备份选择	无	有	有	有	有	有
读性能	高	低	高	高	高	高
随机写性能	高	低	低	一般	低	一般
连续写性能	高	低	低	低	低	一般
需要磁盘数	$n\geqslant1$	$2n(n\geqslant1)$	$n\geqslant3$	$n\geqslant3$	$n\geqslant4$	$2n(n\geqslant2)\geqslant4$
可用容量	全部	50%	$(n-1)/n$	$(n-1)/n$	$(n-2)/n$	50%

近年来，企业的信息化水平不断发展，数据已经取代计算成为信息计算的中心，信息数

据的安全性就显得至关重要。随着存储技术的持续发展，RAID 技术在成本、性能、数据安全性等诸多方面都将优于其他存储技术，例如磁带库、光盘库等，大多数企业数据中心首选 RAID 作为存储系统。

不同的存储厂商的产品在技术、成本、性能、管理、服务等方面各有优势和不足。用户选择 RAID 的原则是：在成本预算内满足数据存储需求的前提下，选择最优的存储厂商解决方案。因此，首先用户需要对存储需求作深入的调研和分析，并给出成本预算，然后对众多存储厂商的解决方案进行分析和对比，最后选择出一个综合最优的存储方案。其中，存储产品的扩展性和存储厂家的售后服务需要重点考察，存储需求(如容量、性能)可能会不断升级，存储产品发生故障后的维修和支持保障，这些都要未雨绸缪。

任务拓展

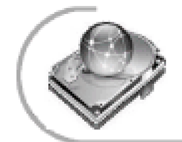
1. 为什么 RAID1 不能替代备份?
2. 调研 RAID6 及其第二校验位的计算。
3. 在备份应用中采用 RAID3 有哪些优势?
4. 讨论不同 RAID 配置对随机和顺序 I/O 的影响。

第5章　虚拟磁带库技术



任务目标

- 了解什么是虚拟磁带库；
- 了解虚拟带库的优缺点；
- 了解虚拟带库的各种技术；
- 了解重复数据删除技术。



项目背景

在虚拟磁带库出现以前，备份软件常用的备份介质是磁带库，其以低廉的成本占据了绝大多数的备份介质市场。但其机械架构产生的大量故障和顺序读写特性大大限制了读性能，这使得人们不断地寻找替代方案。

随着技术的进步，硬盘的成本越来越低，使用硬盘代替磁带成为可能，虚拟磁带库在此时产生。首先，硬盘的随机读写特性使得其读性能大大超过磁带。其次，电子电路架构对比机械架构，故障率也大大减小。

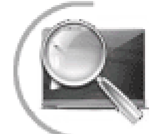
当然，虚拟磁带库和磁带库的差别并不仅限于此，本章会对两者进行进一步的对比介绍。

现在行业上总有虚拟磁带库终将替代磁带库的声音存在，虽然笔者对此并不认同，但不可否认，虚拟磁带库在大量场景中对比磁带库是占据优势的，而且虚拟磁带库的市场占有率也确实在不断增长中。



项目描述

聚比特科技有限公司机房发生了一次重大事故，由于计算机病毒感染，导致机房部分服务器数据大量丢失，给公司造成了巨大的影响。为此，公司领导召集各部门紧急开会，会中针对技术部门，公司领导要求尽快建立灾备系统解决方案，用来应对未来类似数据和应用的安全隐患。



项目分析

聚比特科技有限公司技术部接到建立备份系统的任务，即开始着手准备。首先是确定备份数据存放在哪里，也就是采用什么样的备份介质。通过了解比对，大体选择了三种备选的备份介质：①磁盘阵列；②磁带库；③虚拟磁带库。

磁盘阵列要被使用，需要将其格式化成与使用它的操作系统兼容的文件系统，经历过这

次病毒事件，技术部认为业务系统存储和备份介质采用同一文件系统仍然无法避免数据被同时破坏的风险，所以此方案暂不采用。

磁带库作为备份行业中长期使用的大型备份介质，在安全性上没有问题，但其机械架构故障率较高，后期维护困难，同时也考虑到需要进行数据恢复时用时较长，所以同样被排除在外。

虚拟磁带库同时兼具磁带库的安全性和磁盘阵列的高速性，同时故障率较磁带库要更低，更符合聚比特科技有限公司现在的需要。

所以技术部门正式决定采用虚拟磁带库作为公司备份系统的存储介质。



项目实现

技术部门在决定采用虚拟磁带库后，立即着手虚拟磁带库的选择，发现虚拟磁带库也有各种实现方式和附加功能，为此，技术部对主流虚拟磁带库进行了一次全方位的了解。

5.1 虚拟磁带库介绍及相关产品对比

虚拟磁带库又称为 VTL，是备份软件的磁带驱动或磁带库，其本质是一个硬盘阵列。因此它能以磁盘速度而不是原来缓慢的磁带速度备份和恢复会话。尤其吸引眼球的是，借助 VTL 恢复操作可以直接针对某个文件进行，而无须按部就班地读取整个磁带，因为硬盘是随机访问设备。

虚拟磁带库是一种企业级的数据存储系统，它可以把基于阵列的受 RAID 保护的硬盘驱动器模拟成物理的磁带驱动器，并且将数据备份到其中。使用虚拟磁带库时，备份软件的架构和过程跟真实的带库没有很大区别。虚拟磁带库工作原理跟磁带一样，只是写数据时，把磁盘模拟成磁带。由于虚拟磁带库可以模拟磁带存储系统，任何规模的企业都可以很方便地在其系统中使用硬盘模拟生成的虚拟磁带库了。

VTL 具有两大优点：管理方便、性能良好。实现磁盘到磁盘直接备份需要标准共享磁盘阵列的所有配置步骤。相比之下，如果你告诉 VTL 应该模拟多少虚拟磁带驱动器、多少虚拟盒式磁带，VTL 软件就能自动实现配置，为虚拟磁带合理分配磁盘数。我们就来深入了解虚拟磁带库的组成及利弊。

5.1.1 架构

在考虑 VTL 的优势之前，你需要回答以下问题：

(1) VTL 的扩展性怎么样？能否满足更高的连接需求和容量需求？

(2) VTL 具备哪些管理功能？与备份软件的集成性怎么样？

(3) 压缩、数据重复删除、加密等功能会对 VTL 的性能和容量带来哪些影响？

VTL 有四种部署架构，分别以不同的方法解决上述问题。

(1) VTL 存储产品。这种 VTL 是一种独立的产品，包括磁盘驱动器和控制器，而 VTL 软件寄存在控制器中。具体实例包括 Copen、EMC、Quantum、Network Appliance (NetApp)等公司的产品。

(2) VTL 服务器产品。专用服务器与 VTL 软件同时加载，而且通过光纤通道(FC)与外部磁盘阵列相连。Diligent Technologies、FalconStor 软件公司、Maxxan 系统公司、Neartek、Sun Microsystems 等公司分别提供这种技术。

(3) 基于网格的架构。Sepaton 公司的 S2100-ES2 允许用户给 VTL 添加服务器节点，扩展服务器，从而满足性能或 FC 端口引起的容量需求。

(4) 基于磁带库的架构。ADIC 公司的 Pathlight VX 系列产品、Spectra Logic 公司的 Spectra T950 都允许用户将数据存储在磁带库的磁带或磁盘介质中。

每种架构都有各自的优缺点。VTL 存储产品很容易设置和部署；但是当产品的性能和容量达到极限时，用户需要部署更多的 VTL 产品，从而带来管理问题。VTL 服务器产品可以帮助用户扩展服务器，扩大容量，增强性能，同时又能利用不同供应商的存储；不过 VTL 服务器产品也可能成为性能瓶颈，引入未经测试的配置方式。Sepaton 公司的 S2100-ES2 是基于网格的 VTL 架构，当容量、连接、性能等都达到临界值时，S2100-ES2 能方便地扩展，但最终还是会和存储产品一样再次面临限制问题。ADIC 公司的 Pathlight VX 系列产品、Spectra Logic 公司的 Spectra T950 用一种框架管理磁盘和磁带，但是会使得由 VTL 创建和导出的磁带的管理工作非常依赖于 VTL。

5.1.2　各种 VTL 产品间的重要差异

购买虚拟磁带库(VTL)时，首先要了解各项可用的功能，这样才能创建你所需的功能清单。将供应商的功能与你的需求一一对应，然后就能列举出暂作考虑的产品。供应商的功能可分为“必需的”“最好拥有的”以及“不必要的”。如果功能不是“必需的”，就不应该影响决策过程，当然还应该考虑到价格因素。

1. 主框架及开放系统 VTL

选择 VTL 系统的首要标准应为：你的环境是需要主框架还是开放系统的 VTL。在主框架和开放系统环境中，VTL 的使用方法截然不同，因此，两种 VTL 的设计自然也不同。主框架 VTL 旨在利用磁盘存储，加快磁带数据的访问速度。在数据处理过程中，数据在磁盘和物理磁带之间动态迁移。相反，开放系统 VTL 通常不会处理磁带中的数据；而是利用磁带实现备份和归档。

主框架 VTL 的供应商包括：Diligent Technologies、IBM、Sun Microsystems 及其他一些小公司。过去 40 年，虚拟存储技术渐渐呈现。但是，直到 20 世纪 90 年代后期，IBM 和 StorageTek(现属于 Sun 公司)推出主框架 VTL，虚拟存储技术才真正为人所知。IBM 和 Sun/StorageTek 公司的主框架 VTL 系统在大型的终端磁带库之前，使用了相对较小的磁盘缓存。

主框架 VTL 出现几年之后，开放系统 VTL 也随之产生。目前，开放系统 VTL 的初始厂家有三家——FalconStor Software、Quantum 和 Sepaton 公司，他们再把软件转给其他供应商。这三家厂商同时通过增值分销商和零售商直接或间接提供打包产品。

其他一些开放系统供应商只通过直接和间接渠道交付 VTL 产品，包括 Data Domain、Diligent Technologies(被 IBM 收购)和 NetApp 公司。IBM 和 Sun 公司均生产主框架 VTL，并直接或通过渠道交付 VTL。Diligent 也拥有主框架 VTL，可通过 Hitachi Data Systems(HDS)获得。

2. VTL 的关键影响因素

在选择 VTL 时，可扩展性是一项重要的考虑因素，其衡量标准为容量、性能、仿真引擎的数量、端口数以及相关的项目。存储容量的范围很广，端口数和主机链路也是如此。重复数据删除技术对容量的影响非常重大，可以将系统的有效容量或虚拟容量提高几倍。

仿真引擎或数据迁移器的数量可为 1～8 个，有时候甚至达到 16 个。但是，大型的多引擎系统可能无法在多个元素之间实现双主动故障转移功能，也无法综合管理多个系统。

重复数据删除技术是 VTL 最新拥有的一项功能，主要有两种实现方式，各有优势。重复数据删除功能可以在线执行，将数据流化；也可以在存储数据后再加以执行。在线方式无须为尚未删除的数据提供额外空间，因此能提高空间利用效率。第二种方式需要的空间较多，但是吞吐速率较高。目前的产品一般都采用其中的一种方法，用户在购买 VTL 时需自行选择。如果你的备份要求是减少备份时间，那么第二种方法相对合适；如果你的部署要求是离线迁移数据，以通过 VTL 复制迅速恢复灾难，或者最重要的目标是节省空间，那么在线方式是最好的选择。

在概念上，压缩与重复数据删除技术类似，只是压缩技术出现较早。几乎所有的供应商都提供软件压缩功能以及硬件加速功能。通常，软件压缩功能会减慢整个数据的备份过程。磁带驱动器种类不同，压缩类型也不同。如果 VTL 仿真磁带驱动压缩功能有别于物理磁带，那么将虚拟磁带输出到物理磁带就会花费多盘磁带，浪费磁带匣的空间，甚至发生操作故障。因此，匹配虚拟压缩算法和物理算法非常重要。

另外一种方法是支持托管备份程序，即将备份服务器嵌入 VTL 产品中。这种方法的好处在于无须外部 I/O 活动或备份主机，就能对物理磁带复制过程实现虚拟化。有了托管程序，就无须外部 SAN、存储或备份主机活动，这样就可以避免与主程序争夺线路。

磁带缓存和支持外部磁带功能关系到物理磁带介质如何与 VTL 产品实现集成。如果支持外部磁带，具有磁带驱动器的物理库就能直接与 VTL 相连，这样就可以离线复制或输出物理磁带，或者实现长期存储。而磁带缓存的概念来源于主框架 VTL 系统，是指数据首先流入 VTL，作为缓存，随后继续流入附属的物理磁带。这样，就不必复制或迁移数据到终端磁带中。相反，当 VTL 磁盘缓存充满后，数据会自动流入磁带。

VTL 的复制功能是一项非常重要的功能，其形式取决于 VTL 的利用方式。如果客户想要替换磁带，并保持灾难恢复功能，那么 VTL 复制功能必不可少。有些环境利用 VTL 加快备份和恢复过程，但是仍然利用物理磁带实现离线灾难恢复，这时就不需要 VTL 复制功能。

相比预先配置的产品，客户定制的 VTL 方案能更好地满足特定数据中心的 VTL 需求。利用定制的 VTL 方案，客户可以更加灵活地部署与主供应商具有相同 VTL 功能的系统。系统集成商、增值分销商等能提供为客户定制的 VTL 解决方案，满足客户独一无二的需求。存在许多 VTL 产品、VTL 代理商、增值分销商可供选择，不过与 VTL 技术的选择范围相同。

了解各种可选产品和注意事项后，我们就能编制一份比较清单，列举你的具体需求以及几家供应商的产品。这么做有助于评估 VTL 产品间的重要差异，找到最适合的解决方案。

5.1.3　虚拟磁带库和磁带库的对比

通常我们把物理磁带库称作磁带库。广义的磁带库产品包括自动加载磁带机和磁带库。自动加载磁带机和磁带库实际上是将磁带和磁带机有机结合组成的。自动加载磁带机是一个位于单机中的磁带驱动器和自动磁带更换装置,它可以从装有多盘磁带的磁带匣中拾取磁带并放入驱动器中,或执行相反的过程。它可以备份100～200GB或者更多的数据。自动加载磁带机能够支持例行备份过程,自动为每日的备份工作装载新的磁带。一个拥有工作组服务器的小公司或分理处可以使用自动加载磁带机来自动完成备份工作。

磁带库是像自动加载磁带机一样的基于磁带的备份系统,磁带库由多个驱动器、多个槽、机械手臂组成,并可由机械手臂自动实现磁带的拆卸和装填。它能够提供同样的基本自动备份和数据恢复功能,但同时具有更先进的技术特点。它可以多个驱动器并行工作,也可以几个驱动器指向不同的服务器来做备份,存储容量达到PB(1PB=一百万GB)级,可实现连续备份、自动搜索磁带等功能,并可在管理软件的支持下实现智能恢复、实时监控和统计,是集中式网络数据备份的主要设备。

磁带库不仅数据存储量大得多,而且在备份效率和人工占用方面拥有无可比拟的优势。在网络系统中,磁带库通过SAN(Storage Area Network,存储局域网络)系统可形成网络存储系统,为企业存储提供有力保障,很容易完成远程数据访问、数据存储备份,或通过磁带镜像技术实现多磁带库备份,无疑是数据仓库、ERP等大型网络应用的良好存储设备。

1. 数据读写性能

(1) 物理磁带库:物理磁带库的整体性能由磁带驱动器数量及磁带驱动器支持的标准决定。

(2) 虚拟磁带库:由于采用虚拟化技术,虽然备份软件会发现虚拟磁带库中有若干磁带机,但是执行备份或者数据恢复的时候性能超过了磁带机。因此虚拟磁带库的性能不是由仿真的磁带机标准决定,而是由控制系统和后端的磁盘系统决定。

2. 数据可靠性

(1) 物理磁带库:除非采用磁带Clone的方式,否则由于磁带损坏会导致数据丢失。

(2) 虚拟磁带库:后端采用RAID技术,RAID组中的磁盘损坏一个不会影响虚拟磁带中储存的数据,因此具有更好的安全性。

3. 保密性

(1) 物理磁带库:磁带采用专用数据格式储存数据,并且可以将磁带移出磁带库异地保存,因此具有最高的安全性。

(2) 虚拟磁带库:虚拟带库产品有两类,一类是真正虚拟磁带;另一类是用文件系统中的文件来储存磁带格式的文件。对于第一类产品,除了不能出库以外,具有跟物理磁带库同样的安全性,第二类产品由于存在文件系统,就有可能被病毒或进行人为的破坏。

4. 数据可恢复性

(1) 物理磁带库:磁带存储由于磁带磁粉脱落、黏连、消磁、磁头玷等原因,会在需要恢

复数据时无法获得数据，因此磁带的可恢复性不会是100%。

（2）虚拟磁带库：虚拟磁带库采用磁盘阵列存储数据，有RAID保护，因此可以说虚拟磁带库具有100%的数据可恢复性。

5. 系统可靠性

（1）物理磁带库：由于磁带库中的大量的机械部件，并且要求机械运动精度相当高，所以磁带库本身的系统可靠性就不会很高。

（2）虚拟磁带库：虚拟磁带系统唯一有机械部件的就是磁盘，但是采用了RAID方式进行数据冗余存储，因此虚拟磁带库比物理磁带库的可靠性要高很多。

6. 容量

（1）物理磁带库：物理磁带库的容量非常大。

（2）虚拟磁带库：虚拟磁带库的容量受磁盘系统的容量限制，一般最大不过几十TB。

7. 管理

（1）物理磁带库：物理磁带库基本上都带有图形化管理界面，并且通过一个界面就可以管理整个磁带库系统，包括机械手，磁带机等。

（2）虚拟磁带库：虚拟带库有两类，一类是真正意义上的产品，通过唯一界面管理系统。另外一类属于解决方案，也就是使用专用服务器、操作系统、虚拟磁带库软件和磁盘阵列搭建虚拟磁带库系统。此类方案会造成非常复杂的管理，并且容易引起由于管理不当造成的数据丢失。

8. 软件兼容性

（1）物理磁带库：磁带库基本上各种备份软件都支持，并且很多备份软件的高级特点如Multiplexing、Synthetic Backup都支持物理磁带库。

（2）虚拟磁带库：对于有些虚拟磁带库系统，由于采用黑盒子的管理方式，对于某些备份软件的功能不能很好地支持。

9. 数据迁移

（1）物理磁带库：物理磁带库的数据迁移方式有两种，一种是通过备份软件的磁带复制或者归档功能将一个磁带库中的数据复制到另外一个磁带库中，这种复制与磁带格式无关。第二种就是将介质出库放到另外一个磁带库中。

（2）虚拟磁带库：虚拟磁带库只能够通过备份软件将虚拟磁带中的数据复制到物理磁带库的磁带介质上；并且只有通过备份软件的复制，才能够让备份软件知道数据从何处来、到哪里去，否则进行数据恢复的时候会找不到数据。

5.2 虚拟磁带库的实现方式

目前市场上的虚拟磁带库依照架构不同，大概可以分为三种类型：备份软件型（D2D）、应用服务器型（VTL Appliance）、智能化专用型（Intelligent High Preformance VTL）。

5.2.1 备份软件型虚拟磁带库（第Ⅰ代D2D）

将磁带库模拟软件直接安装在备份服务器上，把备份服务器的某些文件系统分区模拟

成磁带库，从而使备份软件以磁带库方式使用磁盘文件系统。

此类方案下的备份磁盘暴露于主机的操作系统，本质上依然“在线”。在用户看来，依然在线的数据一定是不安全的。举例来说，如果备份服务器不幸被病毒感染，该病毒完全可能在损毁在线磁盘上数据的同时也损毁备份盘阵上的数据。

另外，此类方案占用主机资源，性能受限。

这种方案多由备份管理软件作为一个功能模块提供，价格比较低廉。但由于受制于文件系统，使其应用场合、I/O 性能及数据安全性具有一定的局限。

因此，此类方案主要用于备份缓存，也即先备份到磁盘，然后在服务器不忙时再将备份转移到物理磁带库上，如图 5-1 所示。

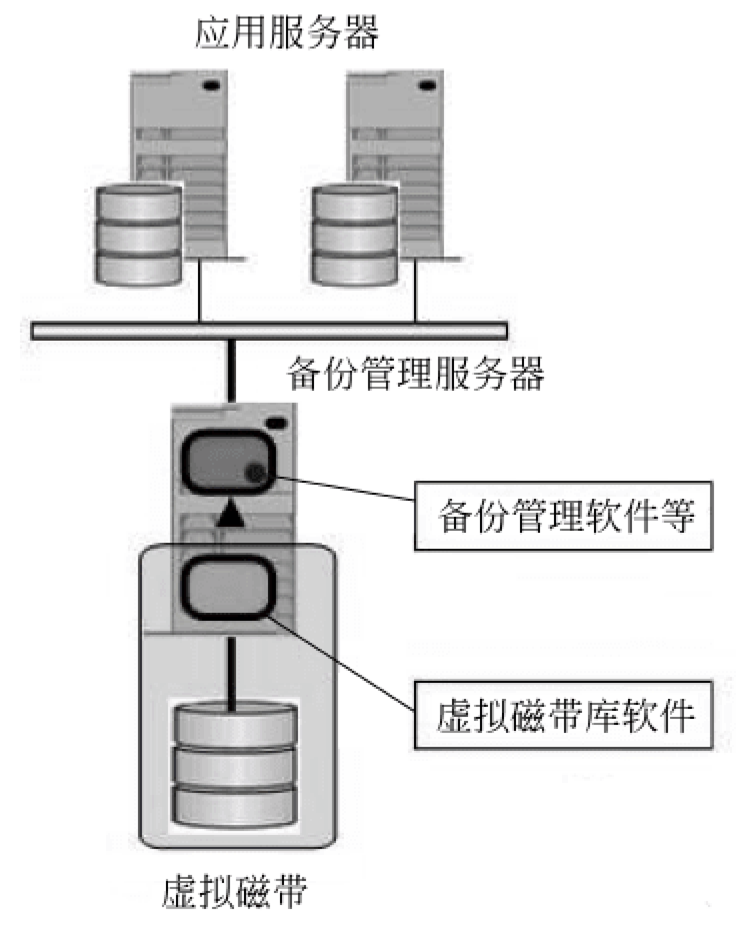


图 5-1　备份软件型虚拟磁带库示意图

5.2.2　应用服务器级虚拟磁带库方案（第Ⅱ代 D2D）

该方案实际上是一种虚拟磁带库的软件实现方案：通过把虚拟磁带库管理软件安装在一台独立的专用服务器（一般是 PC 服务器）内，而将该服务器及所连接的磁盘存储设备模拟成磁带库。

这种方式下，备份服务器或其他应用主机通过 FC 或 SCSI 与专用的服务器连接，此时专用服务器及所连接的磁盘存储系统一起体现为虚拟磁带库（虚拟磁带库）。

与备份软件型虚拟磁带库方案不同的是，备份服务器或应用服务器把专用服务器及其磁盘阵列当作了一台磁带库设备，实现了虚拟磁带库设备与主机设备的物理和逻辑上的分离。主机对这种方案下的虚拟磁带库的读写方式是数据块级（Block-Level）读写，比备份软件类型的读写速度快，并且不会从主机方对备份数据产生误删除操作，主机上的病毒也不会影响备份数据。

此类方案下，虚拟磁带介质磁盘逻辑卷不再是操作系统格式化的扇区，而是和磁带一样的裸介质（raw disk）；其上备份数据也是按顺序逐个字节存放的，在物理层上实现了磁盘读写的线性化，避免了文件系统的碎块问题，充分利用了磁盘设备的高速 I/O 性能。

这种方案的不足是需要利用一台具有一定扩充能力的 PC 服务器作为虚拟磁带库管理器，系统优化性略低，控制器部分采用 PC 服务器结构，不够精简。另外 PC 服务器以及其连接的磁盘阵列管理不统一，不是一体化结构，还容易产生 PC 服务器和后端存储的不兼容问题。如图 5-2 所示。

5.2.3　智能化专用型虚拟磁带库设备方案（第Ⅲ代 D2D）

就存储市场而言，我们熟知的主流磁盘阵列就是采用 ARM 结构、嵌入式实时系统作为核心的，最早的磁盘阵列形态是 PC 服务器＋JBOD 磁盘柜，随着市场和技术的不断发展，这种老式的设备在可靠性、性能上都无法与采用 ARM 结构的主流磁盘阵列相比，因此，市场上已经很难看到这种早期形态的磁盘阵列产品了。

就虚拟磁带库这种应用来说，它的发展趋势肯定也会和磁盘阵列一样，因此，基于

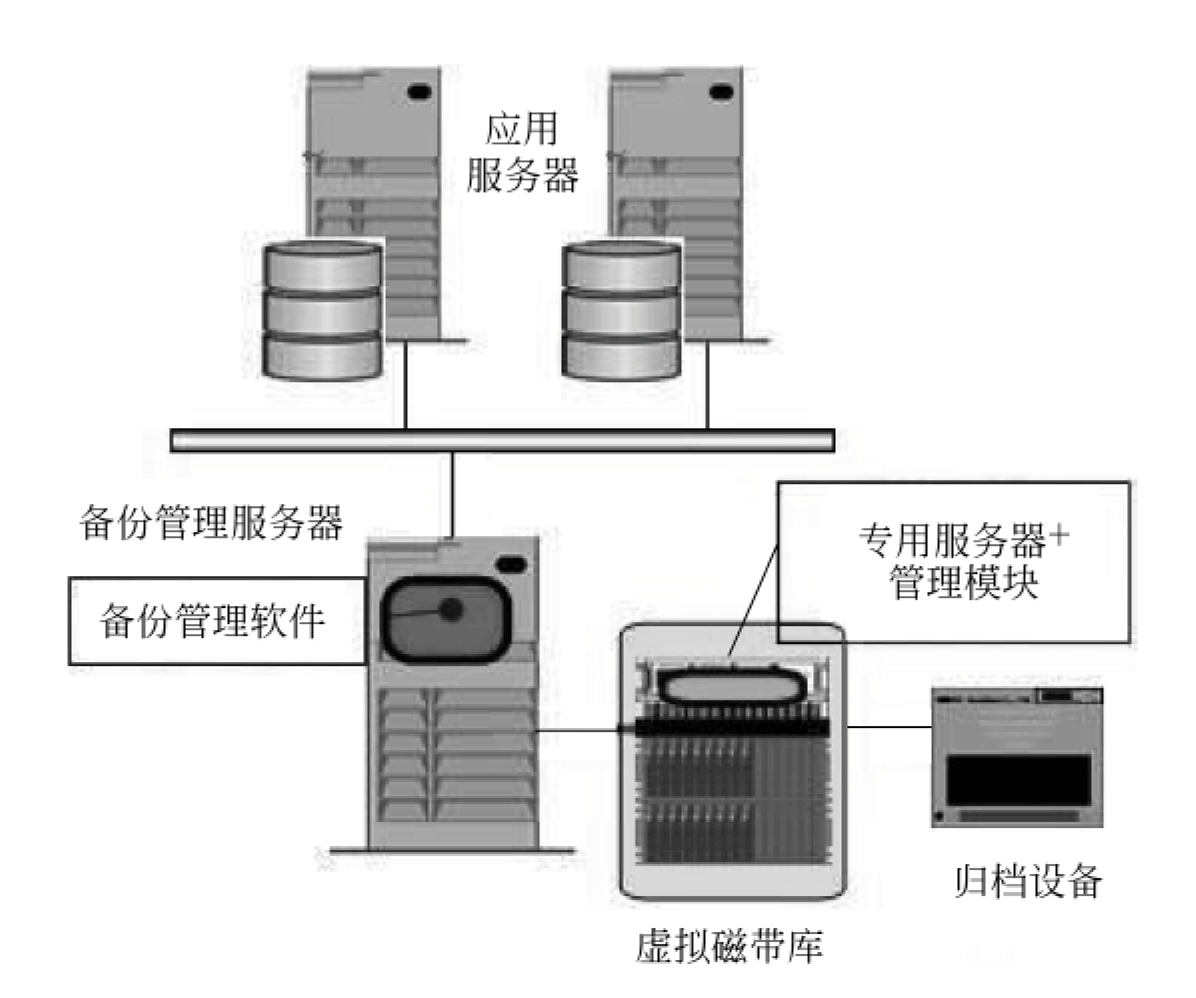


图 5-2　应用服务器级虚拟磁带库方案

ARM 结构的嵌入式系统采用统一化管理，成为智能化专用型虚拟磁带库的基本特征，它将磁带库模拟管理软件固化在特别设计的 ARM 结构、嵌入式实时系统中，就形成了专用的虚拟磁带库设备，这种设备可以配置一定数量和类型的主机接口和后端存储磁盘阵列接口，有的专用虚拟磁带库设备还配置了归档磁带库接口。专用的虚拟磁带库设备硬件结构不同于 PC 服务器，其性能、可靠性比第Ⅱ代应用服务器型虚拟磁带库有了非常大的提升。在设计上采用了精简的硬件模块和精简的操作系统内核(一般为 Linux 内核)，并且充分考虑了与主机及存储设备的连接能力。

专用的虚拟磁带库设备标志着虚拟磁带库技术终于突破了操作系统和 PC 服务器架构的限制，使虚拟磁带库真正成为一种独立的外设，并且真正将虚拟磁带库控制系统和磁盘存储阵列统一管理起来，其使用方式也更接近普通磁带库，而其优越性能也体现得更加充分。如图 5-3 所示。

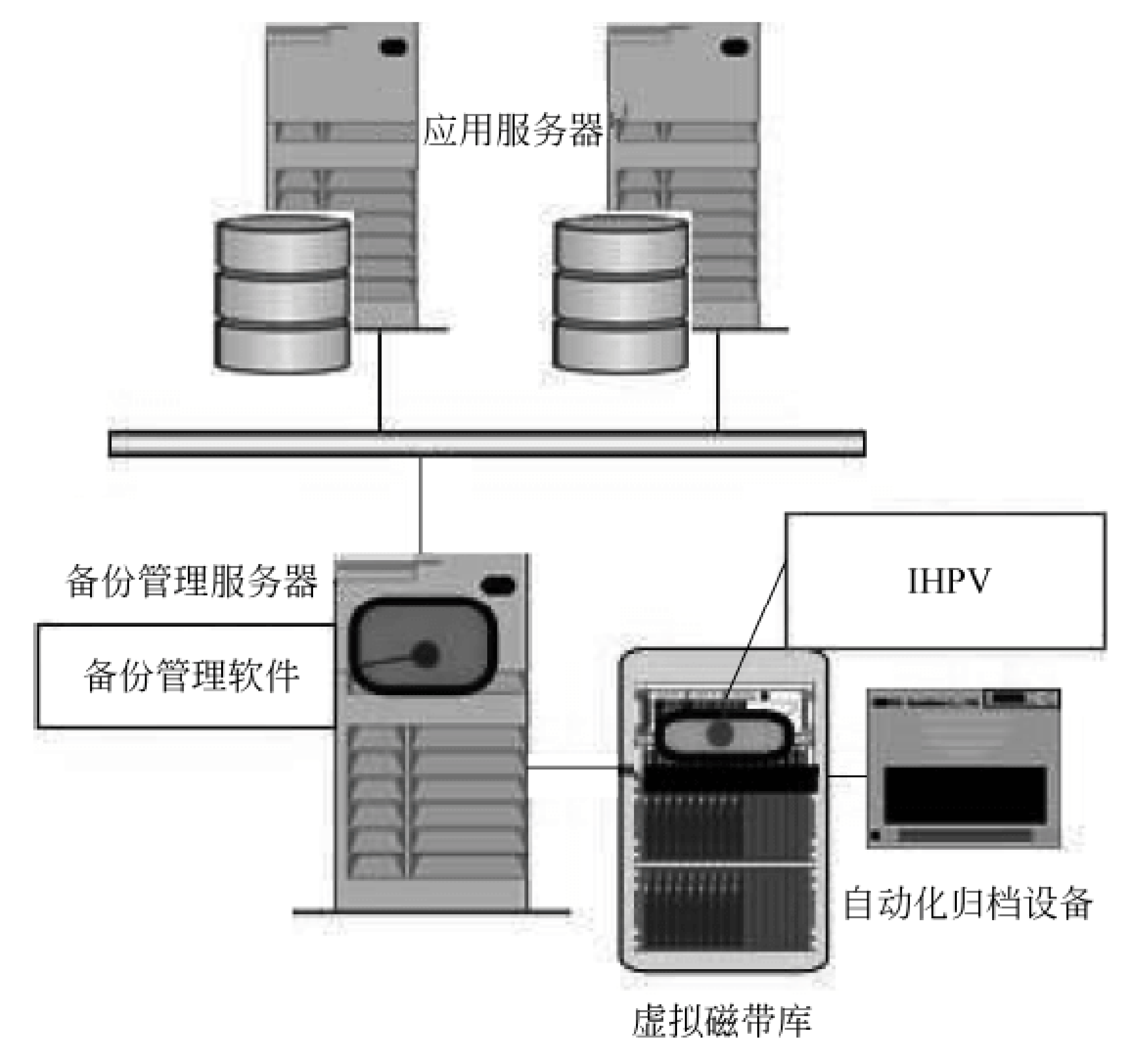


图 5-3　智能化专用型虚拟磁带库设备方案

专用虚拟磁带库设备方案具有如下特点。

- 设备一体化、管理统一化。
- 性能大幅提高。可支持接近磁盘阵列极限速度的备份/恢复速度。
- 免疫病毒。数据安全性等同普通磁带库。
- 避免磁盘碎片。保障性能持续性。
- 兼容性好。标准 FC、SCSI 或 iSCSI 接口设备,兼容流行的主机设备和操作系统。
- 实用性好。与现有磁带库应用方式一致,不用更改现有存储应用软件的管理策略,保护用户投资。

5.3 虚拟磁带库功能介绍

虚拟磁带库(Virtual Tape Library,VTL)的本体是磁盘阵列,可利用磁盘可随机读写的特性,有效提高还原性能,因此 VTL 适于保存还原概率较高的近期生产数据,也就是作为后端真实磁带的缓冲(Buffer),以便缩短备份窗口,所有备份数据将先写进速度较快的 VTL,然后再于其他时间转存到真实磁带上。而磁带属于抽取式媒体,易于扩充容量(只需购买新的磁带),单位成本低,适合用于数据的长期保存。两者如能相辅相成,即可兼顾效率与成本的需求。

5.3.1 新型环境与创新存储的应用

VTL 可利用仿真 SCSI 指令的方式,将磁盘仿真成磁带设备,从而在不改变原有磁带备份架构的前提下,以磁盘取代磁带,达到缩短备份窗口、提高备份成功率、加速还原时间、读取归档数据效率等优势。

对前端的备份服务器或应用服务器来说,VTL 和真实磁带设备是完全一样的,前端完全不会知道后端的"磁带",实际上是 VTL 系统以磁盘仿真出来的空间,因此 VTL 能相当程度地替代真实磁带的作用。

然而随着信息管理法规的要求,长期存档与法规稽核的需求日益增加,磁盘作为长期归档媒体仍有成本过高的问题,相对地,单位成本更低的磁带,更适合这种环境的使用,所以真实磁带设备在现今企业环境中仍占有一席之地。实际上,通常是由 VTL 与真实磁带构成三层式的备份架构,也就是所谓的磁盘到磁盘再到磁带(D2D2T),最前端是在线应用服务器的高性能磁盘,最末端是真实磁带设备,而采用中低价位 SATA 磁盘构成底层的 VTL 则介于两者之间。

5.3.2 直接磁带输出

传统上要将数据从 VTL 转存到磁带上,都必须通过前端的备份服务器来执行,因此会占用备份服务器的作业时间与处理资源。当备份服务器执行 VTL 转存真实磁带作业时,就没办法执行原来的正常备份工作了。

因此新一代 VTL 可支持"直接磁带输出"功能。直接从 VTL 把数据转存到后端真

实磁带上，不会占用前端备份服务器的资源，也不会干扰到备份服务器原来排定的备份工作。

5.3.3 删除重复数据

这项技术可通过对原始数据的分解与特征值演算，除去数据中的冗余部分，让有限的存储空间发挥出数倍甚至数十倍的存储效率，大幅提高磁盘有效的存储容量。

对 VTL 来说，由于可容纳的数据量提高，因此使用者原来每隔几天或每周就需要执行一次的转存磁带作业，可改为间隔 1 个月甚至是半年之久。由于留存在硬盘上的数据量增多，也提高了数据还原或检索的速度。

5.3.4 I/O 负载平衡

传统 VTL 仿真出来的磁带均位于磁盘上的固定位置，某些情况下，可能会发生许多的备份数据流均集中到一个磁盘的问题，以致形成性能瓶颈。

I/O 负载平衡技术则采用动态的 I/O 分配，虚拟磁带并不位于固定的位置，系统可以自动分配备份数据流写入磁盘阵列的位置，备份数据流可动态地平均分布在多个磁盘 LUN 中，避免形成磁盘的 LUN 瓶颈。

5.3.5 硬件压缩

压缩是磁带机必备的基本功能，仿真真实磁带设备的 VTL 自然也须提供这项功能。但软件压缩会造成处理器的负担，因此也降低了 VTL 的整体性能。某些情况下启用软件压缩甚至会使 VTL 吞吐量降低一半，若对不易压缩的离散数据启用软件压缩，那么 VTL 的吞吐量甚至会下降 85%。导入专用的压缩芯片/卡，可在不影响处理器性能的情况下执行压缩工作，比起使用软件压缩的 VTL，吞吐量可高出数倍。

5.3.6 无服务器磁带备份

传统磁带设备或 VTL 的作业控制都需依赖备份服务器，因此会对备份服务器造成负担，使得备份服务器形成备份作业的性能瓶颈。而无服务器磁带备份则不占用备份服务器资源，直接将数据备份到磁带或 VTL 上。如 NDMP 协议在备份中的应用，就是一种典型的无服务器磁带备份。支持 NDMP 协议的存储设备，可直接将数据输出到后端的磁带设备或 VTL 上，而无须备份服务器介入。

5.3.7 销毁虚拟磁带

真实磁带可通过消磁等方式彻底清除数据，而 VTL 的本体是磁盘，因此在淘汰磁盘或是更新设备时，若要确保留在旧磁盘中的数据不致遭到非法复原，除了以物理方法销毁外，必须通过符合美国国防部档案销毁标准 DoD 5220-22M 的工具，通过多次随机数清除与磁盘复写动作，彻底清除 VTL 中保存的数据，这种作业就称为虚拟磁带销毁，可依需要设定销毁特定虚拟磁带数据，而无须更改整个系统。

5.3.8　磁带整合

一种磁带数据转存作业，将多卷存放数据量小的磁带，合并成单卷数据的磁带输出，借此节省媒体与管理成本。除了节省数据占用的磁带数目外，也能在不同规格磁带间转移数据。过去这类作业多半由备份软件负责，但许多新型 VTL 也已内建这项功能，如可仿真成小容量的虚拟磁带来执行前端的备份工作，而在需要转存成真实磁带时，则先转换成大容量的磁带格式后再行输出。

5.3.9　磁带缓冲

一种 VTL 与真实磁带搭配的作业方式，当备份数据写入 VTL 并成为虚拟磁带后，可一并输出成真实磁带，如此 VTL 就好像是后端真实磁带的快照一样。一旦数据写进 VTL（快照）后，马上就转到真实磁带上，然后把 VTL 清空。也可以让数据保留在 VTL 中一段时间，在保留时间内，如果需要还原数据，可以直接从 VTL 中读取，而不用去找磁带，所以还原时也更方便。某些厂商把这功能称为影子磁带（Shadow Tapes）。

5.3.10　按需扩容

VTL 须事先在磁盘区规划出一定数量的虚拟磁带，规划的磁带数量越多，占用空间越大，如果一段时间内没把这些虚拟磁带空间用满，就会形成浪费。而按需分配容量技术切出来的虚拟磁带容量是“虚拟”的，直到数据实际写到那卷虚拟磁带对应的磁盘区后，系统才会实际配给容量；接下来可以视写入数据量的多少，逐次分配对应的磁盘容量给虚拟磁带使用，可大幅提高容量利用率，减少浪费。

5.4　虚拟磁带库的优势与劣势

虚拟磁带库（VTL）是将磁盘当作磁带处理，与磁盘到磁盘直接备份相比，VTL 具有两大优点：管理方便，性能良好。实现磁盘到磁盘直接备份需要标准共享磁盘阵列的所有配置步骤。相比之下，如果你告诉 VTL 应该模拟多少虚拟磁带驱动器、多少虚拟盒式磁带，VTL 软件就能自动实现配置，为虚拟磁带合理分配磁盘数。

如果需要扩展 VTL（并非所有的 VTL 都能扩展），你只需简单地连接辅助存储器，告诉 VTL 存储所在位置，VTL 就能自动利用新分配的存储。既无须运行卷管理器，也无须管理 RAID 群组。

VTL 还有一个非常重要的管理优势，即很容易在多个服务器和应用程序之间共享 VTL。为了使运行同一软件的多个备份服务器能够共享 VTL，应利用内置的磁带库共享性能，大多数商业备份产品都具备这种性能。为了使运行不同程序的多个服务器能够共享 VTL，将一个 VTL 分割成几个小型的 VTL，同时为每个 VTL 分配一定量的虚拟磁带，使每个 VTL 都与各自的备份服务器关联。这两种情况都比较容易实现，而在多个备份服务器之间共享磁盘到磁盘的备份则比较困难。

5.4.1 VTL 的优点

为了理解 VTL 的性能优势,先思考一下备份程序如何将数据写入磁带。通常,在到达磁带的物理结束标记(PEOT)之前,备份程序会源源不断地将数据写入磁带。即使之前写入的一些数据已经过期,备份程序还是会将其添加到磁带中。一旦备份程序达到 PEOT,磁带就写满了。大多数备份程序会将数据完全留在磁盘中,直至所有的备份都到期;然后整个磁盘就到期了,得从头开始写。另外一个备份程序在旁等候,直到磁带中过期的备份数据达到一定数量,随后将没有过期的数据迁移到另外一盘磁带中,“回收”第一盘磁带。此时,第一盘磁带已过期,可以被覆盖。但是,磁带中必须有一部分不被覆盖。

备份程序将数据写入文件系统的原理与此不同。程序会提示操作系统数据需写入特定的某个文件,然后开始将数据写入那个文件。备份都有专属文件,文件过期后,备份就会被删除。备份程序并不了解数据如何写入磁盘。实际上,文件的字节在磁盘中都呈碎片状,从而使得备份性能降低。

VTL 将磁盘当作磁带处理,将备份写入磁盘的相邻区域中,从而使字节不再呈碎片状。在备份程序覆盖磁带之前,分配给磁带的块始终不变,这样 VTL 就能将数据重新写入磁盘的相邻区域——就像数据写入磁带一样。VTL 供应商控制 RAID 卷,必须保证给定的 RAID 群组只能写入特定的虚拟磁带。如果应用程序只是进行读操作或写操作,磁盘的性能会更好。这种差异就能解释为什么最快的文件系统每秒能进行百兆次写操作,而最快的 VTL 每秒能进行千兆次写操作。

VTL 还有其他优点。VTL 能与现有的备份软件、进程、程序兼容。换句话说,一切都如物理磁带库(PTL)般工作。不过例外情况并不是指磁带到磁带直接备份,在 D2D 中,备份软件的工作原理截然不同。

5.4.2 VTL 的缺点

大多数存储管理员认为,VTL 的缺点在于价格昂贵。他们认为如果磁盘阵列需要花费 x 美元,那么使磁盘阵列成为 VTL 就要花费 $x+y$ 美元。但是,y 美元在各个供应商之间又不相同。大多数 VTL 的几个根据容量确定价格,也就是说成本为 x 美元/GB。但也有供应商根据吞吐量制定价格,因此价格由光纤通道(FC)链路的数量确定。VTL 的实际价格从 4～12 美元/GB。所以一味地认为 VTL 比磁盘设备成本高其实是一种误解。

另外,还有个问题就是备份软件的许可费用较高。如果 VTL 和现有的磁带库并列使用,VTL 可能需要辅助磁带库的许可费,支付实际并不存在的磁带库。这就使得 VTL 成本提高。支付费用由 VTL 的配置方式、备份软件对磁带库的收费方式确定。一些备份软件产品对所有的磁带库收取相同的许可费,而另一些产品则根据槽位或驱动器的数量收取费用。在决定如何配置 VTL 的时候,应该考虑到备份软件如何收取磁带库费用。比较 VTL 和磁带到磁带直接备份时,需要记住:备份软件开始对磁带到磁带直接备份收取费用。

但是,备份软件供应商开始根据容量制定价格,努力使 VTL 变得更加友好,许可费等问题将不复存在。

具备压缩功能的 VTL 都是利用带内软件的压缩性能,从而节约空间,但是这就导致性

能降低 50%。如果备份速度受制于客户端或网络,你可能就不会看到这种性能的影响。但是,在本地或没有 LAN 的备份中,备份速度更容易受备份设备的影响。一些供应商选择事后执行压缩功能,这样就既实现了压缩,又不降低性能。

5.5　虚拟磁带库的管理方式

随着磁带库成为近期数据恢复的首要备份目标,存储管理员正在使用一些新方法来充分利用磁带优势:大容量、低成本、可移动性。磁盘是实现快速备份和恢复的最佳介质,但是对于需长期存储和保留的数据,磁带才是最优选择。许多公司在备份过程中使用磁盘和磁带,但是管理这两种介质,难免产生一些问题,比如:

- 磁盘和磁带库的压缩功能在算法上存在差异;
- 已备份数据的加密和解密问题;
- 如何将数据保留在特定介质中,以满足应用程序恢复点目标和恢复时间目标;
- 如何管理虚拟磁带和物理磁带之间的规模差异;
- 根据节约电力或能源成本的原则,优化数据位置;
- 在将数据迁移到磁带之前,重构已经被重复删除的数据;
- 安排数据迁移时间表,使其不影响备份窗口;
- 更新和保护备份软件目录。

尽管一些虚拟磁带库(VTL)能够管理数据从磁盘到磁带的迁移过程,大多数 VTL 是将数据留在备份软件中,初始化数据迁移过程跟踪数据的存储位置。

管理主框架环境中的磁盘和磁带中所引入的原则也适用于开源系统 VTL,无须主框架成本,也没有主框架限制。一些开源系统 VTL 中包含数据管理软件,能够管理数据从磁盘到磁带、磁带到磁盘的复制过程。实现这项功能有两种可用的软件:专业管理软件和第三方备份软件。

5.6　虚拟磁带库数据的迁移

数据存储在 VTL 中后,从 VTL 再将数据复制或迁移到其他介质,实现长期存档或异地数据保护就成为一个难题。从 VTL 迁移或者复制数据通常有三种方法。

(1) 利用 VTL 管理数据在磁带和磁盘之间的迁移活动。

(2) 利用备份软件将 VTL 中的数据迁移到磁带中。

(3) 将数据复制到异地 VTL 中。

将数据从 VTL 迁移到磁带中是最简单的方法,但是,如果将数据迁移或导出到磁带时,VTL 没有通知备份软件目录,备份软件目录就无法保持一致性。有时候,备份软件只了解备份到 VTL 中的数据,而不了解从 VTL 迁移到磁带中的数据,这就是由于 VTL 没有通知备份软件。这时候,在恢复磁带数据之前,就无法恢复 VTL 及其目录。

这也是为什么要采用基于磁带库的架构。一些产品能创建真实磁带，用于导出数据，但是只有在备份软件的控制和导向下才能实现。这样，备份软件目录就能保持一致性，在磁带和磁盘之间迁移数据时不会给备份服务器带来 SAN 流量或开销，从而实现虚拟磁带和真实磁带间的数据复制。其他的 VTL 架构供应商通常会建议：由备份软件管理并实现磁盘和磁带之间的数据迁移。但是，这种方法会降低备份服务器和 SAN 的性能，只有在备份活动不频繁的时期才可以使用，以减少对性能的影响。

从存储节点发送更新消息到主目录，从而使主目录保持一致性。存储节点在虚拟磁带和真实磁带间来回迁移数据，这样就能根据磁带位置更新主备份服务器中的目录。

迁移数据、异地存档数据还有一种方法，就是安装另外一个 VTL，然后在两个 VTL 之间复制数据，很多 VTL 产品都支持用户在两个站点的 VTL 之间异步复制或者迁移数据，几乎不再需要真正的磁带。但是，这种方法不容易扩展，而且需要很大的网络带宽。只有当需要迁移或者复制的数据很少时，你才可以使用这种方法。

新型 VTL 的一些特征大大增加了磁盘中可以存储的数据量，同时也使数据从磁盘到磁带的复制过程变得更加复杂。VTL 的压缩算法不同于目标磁带驱动器使用的算法。因而，管理员在将数据复制到磁带时，可以采用以下三种方法中的一种：

(1) 解压缩 VTL 中的数据，然后在磁带驱动器中重新进行压缩；

(2) 关闭磁带驱动器的压缩功能，直接将数据从磁盘复制到磁带；

(3) 关闭 VTL 中的磁盘压缩功能。

实际上，这些方法都不是太理想。第一种方法在将数据复制到 VTL 和磁带驱动器时，计算机开销增加，不过这种方法可能最易为人们接受，取决于管理员将数据从磁盘复制到磁带所需的时间以及 VTL 的性能。第二种方法能够减少磁带复制过程的性能开销，但是公司必须首先将数据从磁带恢复到 VTL 中，然后再从 VTL 中恢复。关闭 VTL 的压缩功能可能需要 2 倍甚至 3 倍的容量才能存储数据。

VTL 的重复数据删除技术会带来同样的问题。由于磁带驱动器本身并不支持重复数据删除技术，因而如果 VTL 需要使用被删除的重复数据，就必须首先以原始格式重构数据，然后将数据发送到磁带中。这就需要留有足够的时间，确保 VTL 的性能，以便重构被删除的重复数据，然后将其离线发送到磁带中。从技术上讲，被删除的重复数据可以复制到磁带中，但是，这样会使得数据恢复过程依赖于 VTL。

大多数公司不会遇到密码问题，因为一般情况下，只有在将数据离线迁移之前才会对数据加密。那样备份软件和磁带驱动器就只会在将数据存入磁带时才加密数据。

5.7.1 重复数据删除的概念

重复数据删除是基于数据自身的冗余度来检测数据流中的相同数据对象，只传输和存储唯一的数据对象副本，并使用指向唯一数据对象副本的指针替换其他重复副本，相比于传

统的数据压缩技术，重复数据删除技术不仅可以消除文件内的数据冗余，还能消除共享数据集内文件之间的数据冗余。

5.7.2　重复数据删除的技术基础

1. 产生背景

重复数据删除是一项专门用于在减少需要备份的数据量、最大化存储利用率的技术，使更多的备份数据在线保留更长时间。通常来讲，重复数据删除技术会将最新的备份数据与已有的备份数据进行比对，从而消除冗余数据。这项技术的优势在于数据的减少不仅使存储的效率更高、成本更节约，带宽的利用也降到最低，使更经济、更快速的实现备份数据的远程复制成为可能。然而，目前市场中的重复数据删除解决方案还有很大差异，有些解决方案会使备份处理的速度变得很慢，甚至引发无法预计且不可恢复的数据丢失。

2. 技术对比

多年以前的重复数据删除技术，可以看作是文件级的技术，当时称为“单一实例存储”(Single Instance Store，SIS)，通过 SIS 技术，整个文件系统或电子邮件系统的重复文件可以减少到只保留单一的复制，当再出现这一文件时，会指向到这个单一复制，从而减少容量需求。这一技术通常用于电子邮件管理及归档系统。

今天的重复数据删除技术利用了与 SIS 类似的数据缩减概念，但却使之向前迈进了一大步，实现了块级(子文件)重复数据删除。当读取数据时，系统利用 Hash 算法识别唯一的数据块，系统将保留 Hash 索引，每个 Hash 编码指向一个不同的数据块。当新的备份发生时，会自动与现有的块进行比对，如果索引中已经有相同的块，数据将会被删除或被指向块的指针所代替；反之，则会被保存并在索引中为其创建一个新的 Hash 编码。这里提到的唯一数据块的大小，会根据用户选择的重复数据删除解决方案的不同而不同，平均大小在 4～24KB。

与文件级 SIS 技术相比，块级的重复数据删除技术可以说更具优势。它可以在不同的文件或应用中消除冗余数据块。比如，如果一个文件只做了相当小的一点修改并保存，块级重复数据删除技术则将只保存发生变化的数据块。按这种方式，块级重复数据删除可以提供更好的压缩比率，特别是应用于巨大数据量的情况下，如数据库或全备份之后。

下面的例子进一步说明了 SIS 和块级两种重复数据删除技术的结果比较。

一个企业向 1000 个邮件地址发出一封带有注册内容的活动邀请函，传统的备份应用将会把文件备份 1000 次。SIS 解决方案可以识别出文件是相同的，将只会保留一份备份，然后生成 999 个指针指向那个唯一的备份，因此，大概节省了 99.9%的空间。但是，之后这 1000 个参会人可能会将注册内容填好后回复给发件人，由于他们的名字不同，SIS 解决方案就会备份 1000 份回执。

块级解决方案在发出邀请函的时候执行了相同的重复数据删除处理，也实现了 99.9%的存储空间节省。但是，当收到 1000 份参会人的回执时，块级解决方案会发现每个回执的大部分内容与第一封收到的是相同的并已经备份过，因此，它只会对其他 999 封回执的相异数据块(块级)进行备份。这种方法与 SIS 技术相比，则还可以节省额外的 99.9%的存储空间。

3. 优势对比

由于块级重复数据删除技术只保存发生变化的数据，极大地减少了所需的备份容量，使用户不用再像从前一样购置大量磁盘，从而将用户备份系统的总成本降到最低。更少的磁盘致使存储系统更少，电力及制冷需求更小，同时还降低了整体系统的复杂性。

与传统的磁带备份解决方案相比，重复数据删除技术完全改变了磁盘备份的经济性。通过这一技术，更多的用户可以负担得起用磁盘备份取代整个或是部分磁带备份的解决方案。磁盘备份相对于磁带备份来说，提供了更好的备份及恢复性能。通过利用磁盘备份的方式，用户可以进一步改善他们的服务品质协议(Service-Level Agreements，SLAs)。

当通过数据复制实现灾备时，重复数据删除技术也可以带来卓越利益。由于减少了备份数据量，重复数据删除技术将主站点与远程数据中心之间的带宽需求、数据传输成本、复制时间都降到最低。企业利用相对经济的 WAN 网络即可在任意地理空间范围内执行复制操作，实现了额外的成本节约。

4. 使用场景

理论上讲，重复数据删除技术可以用于任意地点存在的数据。它可以与在线或是离线的数据一起工作，可以在文件系统或是数据库中，也可以在其他应用中。总的来说，哪里有大量的重复数据，它就能在哪里呈现出最大的利益回报。

而最好的应用示例是企业级备份。企业大都是每天做一次全备份，两日的全备份中通常只有很小部分的数据是不同的，不会超过 5%，大部分备份扇区都是相似的。这种情况下，重复删除为备份系统带来了最佳的利益回报。因此，大多数重复数据删除解决方案都是专为备份系统而设计的。

5. 发展趋势

从性能的角度看，只能提供单台重复数据删除服务器/设备的解决方案根本无法满足每天需要备份成百 TB 数据或每天需要多次备份的大型企业的需求。在这种情况下，重复数据删除反而会造成瓶颈。一些企业考虑用两台或多台设备完成这一工作，但由于每台设备都保留了它自己的 Hash 索引，设备根本不能鉴别出重复的数据是否已经由另一台设备备份。这种方法不仅会影响到重复数据删除的比率，还会增加维护的工作量。

单台重复数据删除服务器/设备也会成为单点故障。如果重复数据删除服务器/设备发生故障，所有的备份工作则都将失败，更糟的情况是造成恢复过程中的失败。尽管单台重复数据删除服务器/设备在实际应用过程中出现故障的情况非常罕见，但这对企业来说还是一个潜在的、无法接受的风险。

其实，针对这个难题有一种解决办法，就是重复数据删除集群，即利用双工故障切换架构。在集群的重复数据删除架构中，2 台或 4 台重复数据删除设备被虚拟化为一台，为用户提供了双倍或四倍的吞吐及处理能力。如果它们中的任意一台出现故障，另外的设备就会自动地接管它的工作负荷，以确保处理的连续性。由于集群仍然保留的是单个的 Hash 表，因此它不会影响到重复数据删除比率。

5.7.3 重复数据删除技术的分类

基于不同的区分原则，我们可以对重复数据删除技术进行不同的分类，根据进行重复数

据删除操作位置的不同，可以分为源端重复数据删除和目标端懂得重复数据删除，源端重复数据删除可以直接在文件系统内部实现，能够节省网络带宽，但数据缩减率不高，在目标短时间内进行重复数据删除便于采用硬件来实现，可以消除不同源之间的共享数据，获得更好的数据缩减率。

而根据进行重复数据删除操作时机的不同，又可以分为在线重复数据删除和离线重复数据删除。在线重复数据删除能够及时缩减数据，不需要进行后处理，但对吞吐量有影响。离线重复数据删除又称为后处理重复数据删除，是数据存储之后再进行重复数据删除，可以并行地处理，但需要保留足够大的空闲空间。

按进行重复数据删除操作颗粒度的差异，还可以分为文件级、块级、字节比特级重复数据删除。而块级又可以根据划分块的长度是否可变，分为定长块和变长块的重复数据删除技术；重复数据删除操作的粒度越小，删除的冗余数据越多，但是实现的复杂程度和系统开销也相应增加。

此外，根据重复数据删除操作范围的不同，可以分为全局重复数据删除和局部重复数据删除；根据重复数据匹配效果的差异，还可以分为精确重复数据删除和近似重复数据删除。

目前，重复数据删除技术主要应用于备份、容灾和归档系统存储中，这主要是因为备份、容灾和归档应用以写操作为主导，其数据集内具有大量的冗余数据，对于备份和容灾而言，重复数据删除技术不仅能节省空间和优化性能，还能简化数据备份模式，并缩减备份窗口和优化网络带宽，实现高效的数据备份和恢复。而对于归档应用，重复数据删除除了能提供快速的写操作以外，还容易实现数据的不可擦除和不可重写特性，这对归档数据的审计和安全管理具有很大的吸引力。

凭借在数据缩减上的优势，重复数据删除技术也应用在虚拟机环境下的主存储系统中，由于虚拟机环境中的每个虚拟机都要求为其操作系统采用专用的存储，用户有可能为很多虚拟机安装同样的操作系统和应用程序，因此，利用初伏数据删除技术可以为基于虚拟机的主存储系统节省大量的存储空间。

5.7.4　重复数据删除操作的基本原理

（1）数据划分：原文件按照给定的数据划分策略分割成若干个 CHUNK，并为文件建立一个 CHUNK 列表。一般情况下选择的 CHUNK 粒度越小数据缩减率越高，元数据量会相应增加，系统开销也会增大。

（2）特征选择：基于每个 Chunk 的内容期望选择能唯一标识它的特征，并将特征值添加到文件的 CHUNK 列表中。一般选择抗冲突加密 HASH 值作为其特征，如 SHA-1 和 MD5 等算法；有研究者认为 HSAH 函数存在碰撞和生日悖论值得怀疑，特别是在海量数据存储系统中 HASH 冲突不可忽视，一些系统会因为利用基于属性的识别文件或者对相同 HASH 的碰撞而引起数据丢失，如 IBM 公司的 HyperFactor 技术。

（3）相同检测：将 CHUNK 的特征在 CHUNK 索引中进行比对以确定相同数据。随着存储系统的扩展，CHUNK 索引会越来越大，很容易使相同检测成为整个重复数据删除过程中的瓶颈。

（4）冗余消除：通过相同检测，如果发现 CHUNK 索引中有相同的特征，则不需要保存

此 CHUNK。在获取文件时，通过文件的 CHUNK 列表和 CHUNK 索引即可找到相应的 CHUNK 副本。

(5) 数据保存：通过相同检测，若没有发现相同的特征，则将相应的 CHUNK 保存到磁盘上，并将 CHUNK 的特征值和 CHUNK 逻辑块的映射信息保存到 CHUNK 索引中，以便进行相同的检测和文件获取。对比传统的存储系统，重复数据删除系统基于内容寻址，而不是基于文件名寻址。尽管减少了写操作，但是由于增加了重复数据删除的处理过程，比传统存储系统的 I/O 性能要低。由于每次只写新的数据，重复数据删除系统具有顺序写、随机读的特点。另外，因为需要先检测 CHUNK 是否与别的文件共享，重复数据删除系统的删除操作性能低，为方便数据审计和安全管理的需要，通过增加保留锁可以很容易实现数据的不可擦除和不可重写特性，此外重复数据删除系统支持对象存储技术，方便数据的管理。

5.7.5 重复数据删除可选择的方式

目前市场中的重复数据删除解决方案提供了多种删除重复数据的方式，如果想找出最适合用户需求的方式，需要考虑两个关键性因素：在哪里进行重复数据删除(源数据端还是目标端)；在备份处理的哪个阶段删除重复数据删除(在备份过程还是备份之后)。

1. 源数据端

一些重复数据删除产品在源数据端工作，这意味着它们是位于主机或应用服务器上的。这种方式要求为每一台需要进行备份的机器安装重复数据删除系统。当数据传送到备份软件之前即识别重复数据并删除。这种重复数据删除方式具有如下优点。

(1) 数据在网络传输前就减少了，改善了传输性能，节省了网络带宽。

(2) 从扩展性角度讲，整体性能与客户端的数量无关，因此，在进行扩展时不会影响到备份性能。

(3) 由于是在对重复数据做删除处理的同时传输数据，因此在处理过程中不再需要额外的存储设备。

然而，这种处理方式也有以下缺点。

(1) 必须在每一台备份服务器上安装重复数据删除系统。在大型企业环境中，特别是具有分布于各地的分支机构的大型企业，软件的安装和维护工作极其复杂。

(2) 重复数据删除处理运行于应用服务器上，会消耗服务器的运算能力，影响到应用性能。这对于大多数企业来说是不可接受的。

(3) 有些重复数据删除解决方案在设计之初并没有考虑到与用户已经安装的备份软件相集成，因此，用户必须为了使用它而更换备份软件，这便增加了部署解决方案时的工作量及风险，对于那些具有大量归档数据的企业用户来说，这种方案并不可行。

2. 目标端

目标端的重复数据删除处理通常是运行于一台独立于主机或应用服务器之外的设备(服务器或工具)之上。下面是几种典型的目标端重复数据删除方式。

(1) 具备重复数据删除功能的虚拟磁带库(VTL)。

(2) 具备重复数据删除功能的磁盘设备。

(3) 具备重复数据删除功能的备份软件。

值得一提的是，目标端重复数据删除解决方案可以与现有的备份软件相整合，运行于备份服务器上。备份数据先发送到重复数据删除设备上，重复数据删除在后端完成。相对于源数据端方式，它的优势如下。

(1) 企业无须改变其当前的备份系统。

(2) 这种架构无须更换应用服务器。由于它不会消耗应用服务器的运算能力，应用性能不会受到任何影响。

(3) 重复数据删除服务器/设备能够从不同的备份客户端删除所有冗余数据。这对于具有大量应用或数据库服务器的企业来说，是非常经济高效的解决方案。

对于那些不想改变现有备份系统，也不想牺牲应用性能的企业用户来说，目标端重复数据删除无疑是最佳的方式。

3. 在线处理

在线处理的重复数据删除，是指重复数据删除服务器/设备从备份软件开始传输数据，数据还没有写入磁盘之前，即执行重复数据删除。它的重要优势是经济高效，可以减少存储容量的需求。它不需要用于保存还未进行重复数据删除的数据集的临时磁盘空间。

尽管在线处理的重复数据删除减少了备份数据的数量，但处理本身会减慢备份的速度。因为重复数据删除是在写入到磁盘之前进行，重复数据删除处理本身就是一个单点故障。因此，如果企业在需要保留备份数据冗余的时候是不能选择在线处理方式的。

4. 后处理

后处理的重复数据删除，也被称为离线的重复数据删除，是指在备份处理完成后再执行重复数据删除。备份数据先被写入到临时的磁盘空间，之后再开始重复数据删除，最后，将经过重复数据删除的数据复制到末端磁盘上。

后处理方式有一个很小的问题在于它需要额外的磁盘空间来保存全部还未删除重复数据的数据集。但是，低成本的 SATA 磁盘可以用作这部分，因此，对于大多数企业来说，这根本算不上什么问题。

后处理的重复数据删除方式带了诸多优势。

(1) 由于重复数据删除是备份完成后在单独的存储设备上执行，因此完全不会对备份处理造成影响。管理员可以随意制定重复数据删除的进程，无论怎样的频率都可以，性能更加可靠、可控。

(2) 先将备份数据保留在磁盘上进行重复数据删除，企业在需要时则可以更加快速地恢复最近备份的文件和数据。

某种角度上讲，在线处理方式也许更适用于小型企业的需求，而后处理方式对于中型及大型企业来说是更佳的解决方案。

5.7.6　重复数据删除的优势

1. 能够与当前环境相整合

一个高效的重复数据删除解决方案应该对当前 IT 环境的影响/中断越小越好。许多企业都选择利用 VTL 备份来避免影响/中断，以在不改变企业当前备份策略、处理或软件的情况下提升备份质量。因此，基于 VTL 的重复数据删除技术在部署时对环境影响也应该是最小的。它将更多的注意力集中在了备份这个巨大的重复数据存储池上。

基于 VTL 的重复数据删除解决方案通常要求使用专用设备，但这并不影响部署的灵活性。一个充分灵活的重复数据删除解决方案应该即可以以软件包形式提供给用户，也可以提供给用户整体的解决方案（Turnkey Appliance），从而最大限度地使用户的现有资源得以利用。

2. 具备可扩展能力

由于重复数据删除解决方案是用于长期的数据存储的，在容量和性能方面的可扩展能力也是非常重要的考虑因素，而且至少要考虑未来五年甚至更长时间的增长计划。那么，在保证快速访问的前提下，你希望有多少数据保存在磁带上？你需要怎样的数据索引系统呢？

优秀的重复数据删除解决方案提供的架构，无论是在初始部署时，还是面对未来系统的长期增长，都应该能保证最优化（Right-sizing）、最经济的架构规模。集群可以帮助用户满足不断增长的容量需求，即使是多数据增长的环境，也不会降低重复数据删除的效率或系统的性能。

这个架构还为存储库保护的部分提供了故障切换（Failover）功能。

3. 支持分布式应用

重复数据删除技术，不只是能为单个数据中心带来利益，对于具有多个分支机构或多个站点的大型企业来说，它可以让整个企业的分布式应用受益无穷。一个包含复制和多级重复数据删除的解决方案可以将这一技术的优势发挥到极致。

举例来说，一个企业由 1 个总部和 3 个区域代表机构构成，可以在区域代表机构部署一台具备重复数据删除功能的容灾设备，使本地存储及向远程中央站点的复制更为高效。这种解决方案使数据复制到中央站点的带宽需求降到最低，它只不过是用来确定远程的数据是否已经包括在中央的存储库中。所有站点中，只有唯一的数据会被复制到中央站点或是容灾站点，否则所需的带宽就会增大。

4. 能够对存储库提供实时保护

保证对删除重复数据的存储库的访问是非常关键的，因此它不能允许有单点故障发生。一个优秀的重复数据删除解决方案应该包括可以在本地存储故障发生时提供保护的镜像功能，同时也应该具备复制功能以在灾难发生时提供保护。这种解决方案还应该在出现节点故障时具备故障切换能力，即使是一个集群中的多个节点出现故障，企业也必须能够及时恢复数据同时还要保证业务持续运营。

5. 找到最适合的整体解决方案

由于业务应用需要和法律法规的要求，存储的数据量还在不断地增加，重复数据删除也快速上升到至关重要的地位。在大幅消除数据量、削减存储需求、最小化数据保护成本和风险方面，重复数据删除可以说是唯一的应对办法。

尽管重复数据删除技术所带来的利益很多，企业还是应该抵御住不时出现的针对这一技术的大肆炒作。无论是哪种方式，重复数据删除的删除比率都可以根据数据自身的格式和保护策略的不同而发生变化。

为了使重复数据删除技术的利益最大化，企业应该从上面提到的几个标准出发，充分考虑，仔细评估，找到真正适合自己的重复数据删除解决方案，而不应该简单地听信于宣传的重复数据删除比率的理论数值。

5.8　虚拟磁带库的趋势介绍

传统的磁带备份软件的核心功能是连接并管理磁带设备，并以磁带的格式读写数据。而近几年出现的虚拟磁带库（VTL）技术完全改变了备份的应用模式，将数据直接写到磁盘上成了备份软件的主要功能。

虚拟磁带库厂商早期发布的产品都是在模拟流行的光纤通道磁带库的功能，优先考虑与主流备份软件保持兼容性；其次是确保虚拟磁带库不会成为备份过程中的单点故障；此外，提供方便的对虚拟磁带进行远程复制的功能，以实现灾难恢复（DR）的目的。虚拟磁带库厂商还与磁盘存储厂商合作，将虚拟磁带库软件与存储硬件集成在一起作为整体解决方案打包销售，主要是为了屏蔽系统安装配置的复杂性。

目前 VTL 技术已经进入主流用户市场，但是与早期的市场情况相比用户的需求却发生了变化。今天，易于部署和使用、能够改善备份性能并且减少物理磁带介质管理问题等虚拟磁带库特性只能满足用户最基本的需求。所以虚拟磁带库厂商现在需要为主流用户提供更先进的功能，以适应不断变化的市场环境。

5.8.1　变化中的虚拟磁带库市场

改变虚拟磁带库市场环境的因素有很多。首先是因为基础技术更加成熟，VTL 厂商已经能够做到让他们的产品适合各种各样的应用环境，并且能够从客户那里获得满意的评价；其次，今天的虚拟磁带库用户不仅了解 VTL 技术，而且对他们自己在数据方面的需求也十分明确。他们会更加谨慎地看待厂商对性能指标的宣传，例如客户会关注一些具体的问题：怎样配置系统才能实现广告宣传的吞吐量指标？重复数据删除的比率实测能达到多少？

另一个原因是备份解决方案的不断进步。大多数传统备份厂商已经开始支持磁盘到磁盘（D2D）的备份，同时也出现了一批专为 D2D 备份定制的解决方案。今天，备份厂商敏锐地意识到了用户的新需求：①追求更快的备份速度解决备份窗口不足的问题；②通过更快的数据恢复技术提高 RTO（Recovery Time Objectives）指标；③使用容量优化技术减少数据的传输量和存储量。

许多新的备份产品都在力争满足上面提到的这些主要需求，那虚拟磁带库厂商对此是如何反应的呢？他们已经推出一批新的功能来争取那些务实的潜在用户，这无疑是个好消息。

5.8.2　用户的需求

ESG（Enterprise Strategy Group）针对虚拟磁带库的用户做了一份调查，了解他们期望虚拟磁带库解决方案中有哪些特性。调查报告列出了几个用户最关注的功能：重复数据删除和改进虚拟磁带库管理两项高居榜首；紧随其后的是产品扩展能力、数据恢复能力和性能；有些用户关注容灾问题，希望数据能够导出设备之外保存，他们需要在 VTL 后端连接物理磁带设备；最后，VTL 需要支持更多的协议（例如在大型机上使用的 FICON 和 ESCON，支持以太网上的 iSCSI）。

调查显示用户越来越关注 VTL 设备的整体功能。厂商已经开始考虑将这些技术纳入到自己的产品发展蓝图之中。例如，在许多解决方案里多个 VTL 之间实现数据复制已经成为一个基本的功能特性。用户也特别关注这些新功能是否能够在自己的工作流程和系统环境中使用。他们感兴趣的是复制是否可以有多个目标？数据在传输过程中是否可以加密？是否有智能的调度策略（在非繁忙时间）或传输带宽限制功能？哪些技术能够减少数据传输量？

5.8.3　下一代产品的增强特性

1. 容量和性能的可扩展能力

存储容量的扩展能力和性能的扩展能力是排名最高的两项用户需求，而且不分伯仲。某些早期的 VTL 产品在设计上存在瓶颈，它们的架构并没有充分考虑备份应用的特点，也没有估计到可能会出现的性能限制。有时候为了减少虚拟磁带库的压力，用户不得不调整原有的备份进程或后台任务。因此，下一代虚拟磁带库必须能够线性增加吞吐性能，以解决备份时间窗口不足的压力，满足恢复时间指标的要求。对容量的限制问题也是一样，设备必须能够方便地进行磁盘的扩容。

2. 容量优化

当用户将更多的数据备份到磁盘介质并且进行长期保存的时候，容量的优化能力就会变得更加关键，比如压缩功能和重复数据删除，下一代虚拟磁带库在某种程度上必须具备以下的特性，才能应付客户对容量和性能的苛求。这些特性包括：同时在多个虚拟磁带库控制器之间实现数据去重；能够根据系统负载情况关闭去重功能；支持实时处理（联机处理）或批处理模式（后处理）下的数据去重。

3. 集中管理功能

如果用户同时部署了多个虚拟磁带库（单台设备容量限制导致的不断扩容，或一开始就是多台的设计），那么就可能带来一些意想不到的管理问题。集中管理策略和多 VTL 设备的集中管理视图，可以减轻管理的负担，这将是下一阶段 VTL 产品的一个重要的功能特性。

4. 灾难恢复

采用 D2D 备份策略所面临的问题是当遇到系统故障或者灾难事件时，如何保证数据的安全。当前最好的办法是在装置外保存一份备份数据的介质拷贝。通常，VTL 厂商采用“磁带导出”的命令，在物理磁带上做一份虚拟磁带的副本；或者通过让 VTL 直接管理物理磁带库的方法，在备份窗口以外的时间创建介质副本。但这两种方法都不能很好地满足用户要求，因为备份软件无法维护所有的元数据，也无法对副本进行控制。另一种办法是通过备份软件发起虚拟磁带到物理磁带的复制，这意味着备份软件要承担额外的处理任务，数据流也必须经过网络（而这看起来根本没有必要）。下一代虚拟磁带库能够用更有效的方法创建 VTL 介质的可离线副本。利用备份软件厂商提供的 API，比如 Symantec 公司的 Veritas NetBackup 备份产品的 OpenStorage 编程接口，虚拟磁带库能够创建从虚拟磁带到物理磁带或者到第二个 VTL 设备的数据副本，同时还能确保两份介质的管理元数据保持同步。目前的虚拟磁带库产品已经具备了本地到远程在两个 VTL 之间实现复制容灾的功能，下一代虚拟磁带库将在此基础上再前进一步，用户可以根据需要灵活地选择部分数据进行点

对点的复制，而且会支持更多的复制拓扑结构，如一对一、多对一以及双向复制。

5. 对 iSCSI 的支持

随着 iSCSI 性能的改进、万兆以太网络的普及，以及存储厂商不断增加对 iSCSI 的支持，都在推动对以太网接口产品需求的增长。有些用户希望用 IP SAN 来替代 FC SAN，或者作为 FC SAN 的补充，他们就会选择支持 iSCSI 接口的 VTL 产品。

众所周知，企业无论规模大小都要面对灾难的风险，所以必须有可用的容灾数据，并且在发生灾难宕机的情况下能够快速地启用。即使发生的不是毁灭性的灾难，但是如果不加以防范，仍然可能付出沉重的代价。对于容灾这个细分市场，虚拟磁带库也是一个很好的解决方案。

现在，我们兴奋地看到 VTL 厂商正在努力提升自身的技术能力以应对市场挑战，比如将重复数据删除功能纳入到产品之中就是一个明显的迹象。而作为用户，我们就准备好去体验下一代的虚拟磁带库产品好处吧。

任务拓展

1. 什么是虚拟磁带库？
2. 虚拟磁带库的优缺点是什么？
3. 重复数据删除的优势是什么？

第6章　数据灾备与恢复技术



任务目标

- 了解灾备技术的分类；
- 了解各种灾备技术的原理和适用范围；
- 了解各种灾备技术的对比。



项目背景

计算机系统可轻松进行复杂工作中巨大数据量的处理工作，却也同样因其保存有巨大数据量的信息而存在一些数据安全上的隐患。一旦因为这些隐患导致数据丢失，其造成的损失几乎无法通过人力短时间恢复。



项目描述

聚比特科技有限公司机房发生了一次重大事故，由于计算机病毒感染，导致机房部分服务器数据大量丢失，给公司造成了巨大的影响。为此，公司领导召集各部门紧急开会，会中其余不表，针对技术部门，公司领导要求尽快建立灾备系统解决方案，用来应对未来类似数据和应用的安全隐患。



项目分析

聚比特科技有限公司技术部已确定了备份介质，当然是需要采用数据备份软件来铺垫最基础的备份系统。但数据备份仅仅是灾备系统的一种基础方式，灾备系统对于数据和应用的安全还有各个级别和技术的解决方案，应对不同场景下的灾难。如何选用各种灾备技术来切实满足公司当前的需要，又符合公司的需求预算，就需要详细了解各种技术后再进行判断。



项目实现

聚比特科技有限公司技术部了解了主流的灾备技术，发现各种灾备技术都能在自己公司找到适合的配置位置，而且一个比一个好，所以，他们联系了灾备公司说明了情况，并进行了前期询价。然而拿到灾备公司报价后，高昂的价格令技术部大吃一惊，其远远超出了公司的预算。至此，他们才明白更高技术含量的灾备技术同样意味着价格的大幅增长。

在进行了激烈的内部讨论后，聚比特科技有限公司技术部决定先搭建灾备系统的基础架构，满足现在的需求。然后再根据公司未来的发展，逐步增加其他灾备技术，步步为营，稳

扎稳打。

6.1　灾备技术介绍

为实现灾备模式所确定的目标，应采用安全、可靠的灾备技术和方案。目前比较成熟的灾备技术主要分为两类：一类是数据备份技术，另一类是数据复制技术。

数据备份技术是通过专业的备份软件，将生产数据备份成归档数据文件进行保存，需要使用或验证备份数据时，再通过备份软件将数据文件恢复成源数据；基于数据备份技术的灾备方案包括本地备份异地保存方案和远程数据备份方案。

数据复制技术是将生产数据直接复制到灾备中心，当生产中心发生灾难需要切换到灾备中心时，灾备中心的数据直接可用。数据复制技术又可分为基于智能存储设备的复制技术、基于主机的复制技术、基于数据库的复制技术和基于存储虚拟化的复制技术，如图 6-1 所示。

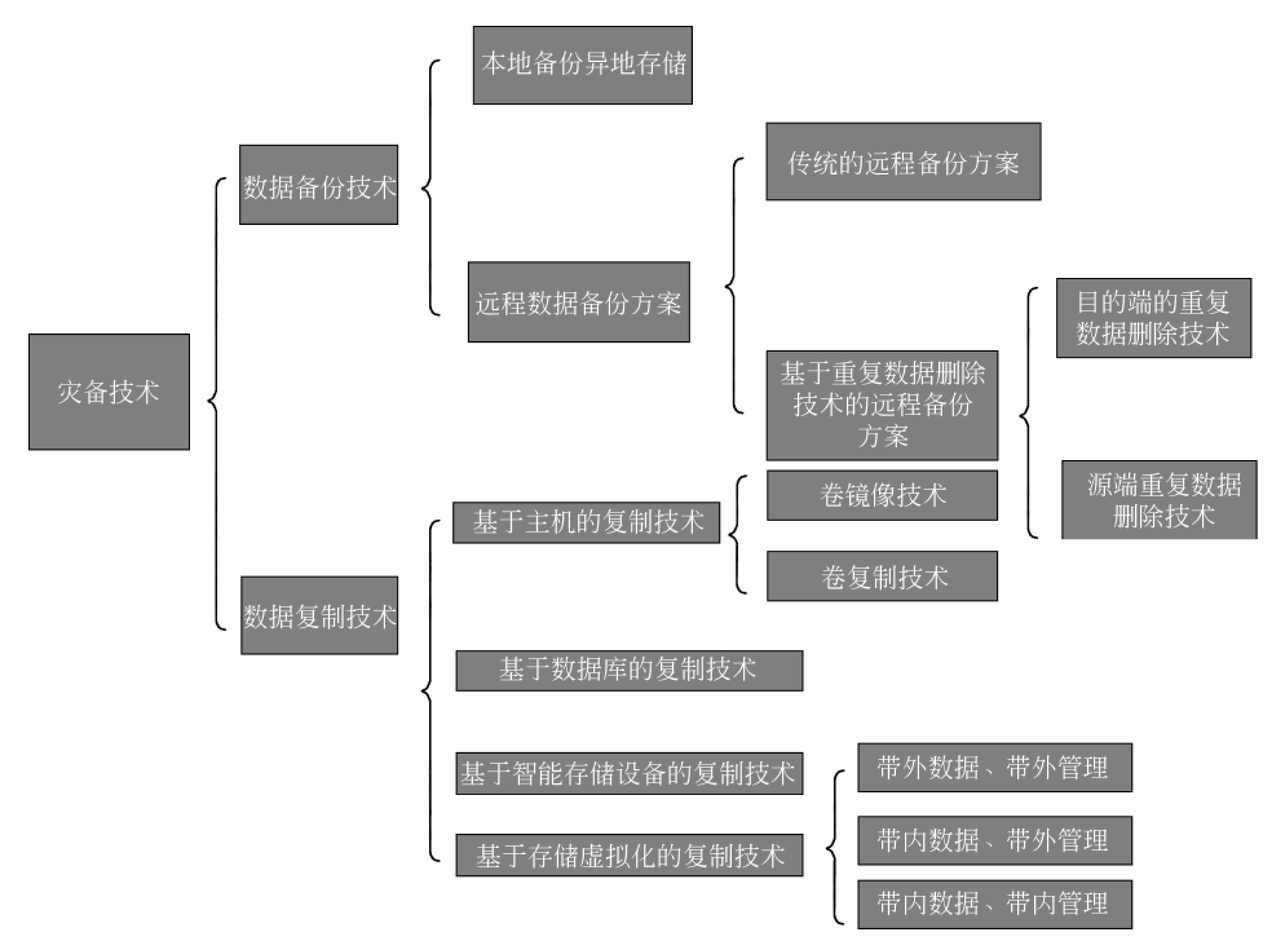


图 6-1　灾备技术分类示意图

由于通过数据备份技术备份的数据不能直接使用，需要恢复后才能使用，因此该技术主要适合于数据级灾备的情况。而通过数据复制技术复制的数据可直接使用，因此该技术既可适合于数据级灾备，也可适合于应用级灾备。

6.1.1 数据备份技术

通过专业的数据备份软件,结合相应的硬件和存储设备,对数据备份进行集中管理,实现自动化的备份、文件归档以及灾难恢复等。在灾难恢复时,需要安装相关系统。基于数据备份技术的灾备方案主要包括两种:一是本地介质备份,异地介质存放方案;二是基于远程数据备份技术的灾备方案。

6.1.2 本地介质备份及异地介质存放方案

1. 技术描述

生产中心通过备份软件按照既定的备份策略将数据备份到本地磁带库上,在通过磁带库备份数据的时候,同时备份两份,一份留在生产中心,一份运送到异址保存,以用于灾难时的数据恢复,如图 6-2 所示。

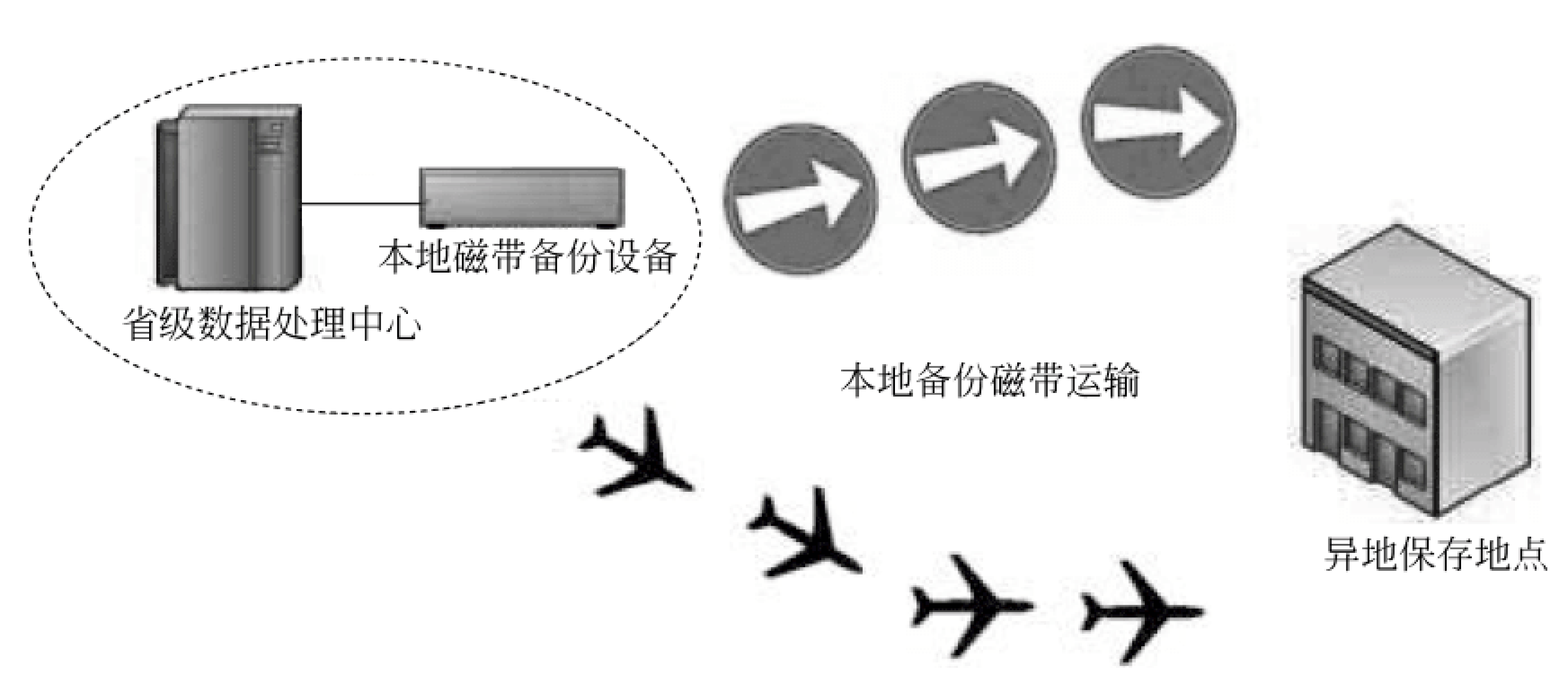


图 6-2 异地备份示意图

2. 资源配置要求

(1) 生产中心需要配置磁带库、备份服务器及备份软件。

(2) 生产中心和异址需要存放保存磁介质的专用柜子。

(3) 将介质定期运送到异地。

3. 方案特点

本方案的优点在于投资小,只需考虑磁带运输成本和异址存放磁带的场地。采用该方案需要同时考虑以下问题。

(1) 该方式的备份过程和恢复过程都较复杂,需要制定严格的流程进行管理。

(2) 磁带通常是在非生产时段进行备份,备份数据时需要考虑延迟。

(3) 在进行数据的恢复操作时必须使用正确的磁带。因此存档磁带的管理工作需要按照专业、规范的流程严格执行。

(4) 在恢复数据库的时候,需按照正确的事件处理顺序执行,以保证能恢复到数据库的当前状况。

4. 适用范围

本方案适用于地市中心生产数据先进行本地磁带备份,再运输到省中心进行磁带保存。

5. 主流产品

目前比较主流的备份软件包括：Symantec Netbackup、IBM TSM、Bakbone NetVault、EMC Networker 等。

6.1.3 远程数据备份方案

远程数据备份方案与本地介质异地存放方案的主要差别在于前者是通过生产中心与灾备中心之间的 IP 网络进行数据远程备份，而后者是通过人为运输的方式将生产中心本地的备份介质传送到灾备中心进行保存。

对于生产中心与灾备中心之间具有 IP 网络连接的情况，可采用远程数据备份方案，本方案适用于地市中心将生产数据通过 IP 专线远程备份到省中心的情况。远程数据备份方案可分为传统远程数据备份方案和基于重复数据删除技术的备份方案。

传统远程备份方案与基于重复数据删除技术的备份方案的主要差别是前者只是将生产数据的备份归档数据传输到灾备中心，由于备份数据传输前没有进行数据优化，因此需占用较大的网络带宽，备份效率较低；而后者是在传输前通过重复数据删除技术识别备份数据的冗余数据段，并将其删除，达到优化效果，经过冗余数据删除后的备份数据量将大大降低，因此可以大大减少对网络带宽的占用。同时重复数据删除技术还可通过加密备份和定期备份验证等方式增加安全性，从而实现了快速、高效、可靠的数据保护。

重复数据删除技术既适用于两点之间的数据备份（如生产中心与灾备中心），也适用于多点之间的数据备份（如多个地市向省中心备份），当添加存储节点时可达到性能的线性增长，不会对重复数据消除效率和系统性能造成影响。同时为保证数据的有效性和完整性，该技术还可以定期进行数据的完整性检验。

根据备份数据处理的对象不同，重复数据删除技术可分为目的端重复数据删除技术和源端重复数据删除技术。

下面介绍传统的远程数据备份方案。

（1）技术描述

传统的远程数据备份方案主要是采用备份管理软件通过生产中心与灾备中心的 IP 网络将生产数据直接备份到灾备中心，其备份效果是生产中心和灾备中心均有生产数据的备份。当生产数据遭到破坏时，可用生产中心的备份数据进行恢复；当生产中心发生灾难时，可用灾备中心的备份数据进行恢复，如图 6-3 所示。

该方案的实现方式是生产中心通过备份管理服务器发起备份操作指令，将生产数据备份到本地物理或虚拟带库上，完成本地备份操作；生产中心备份管理服务器通过 IP 实现对灾备中心备份服务器的作业管理及调度。通过广域网将生产中心物理或虚拟带库上的备份数据传递到灾备中心的物理带库上；灾备中心通过备份软件将数据备份到磁带上，完成备份操作。

（2）资源配置要求

① 在生产中心配置备份管理服务器，并在备份管理服务器上部署备份管理软件，备份管理服务器通过 HBA 卡与光纤交换机进行连接，以保证备份管理服务器能通过 SAN 网访问到存储设备。

② 生产中心可配置虚拟带库，也可配置物理带库。其中虚拟带库可通过两种方式连接

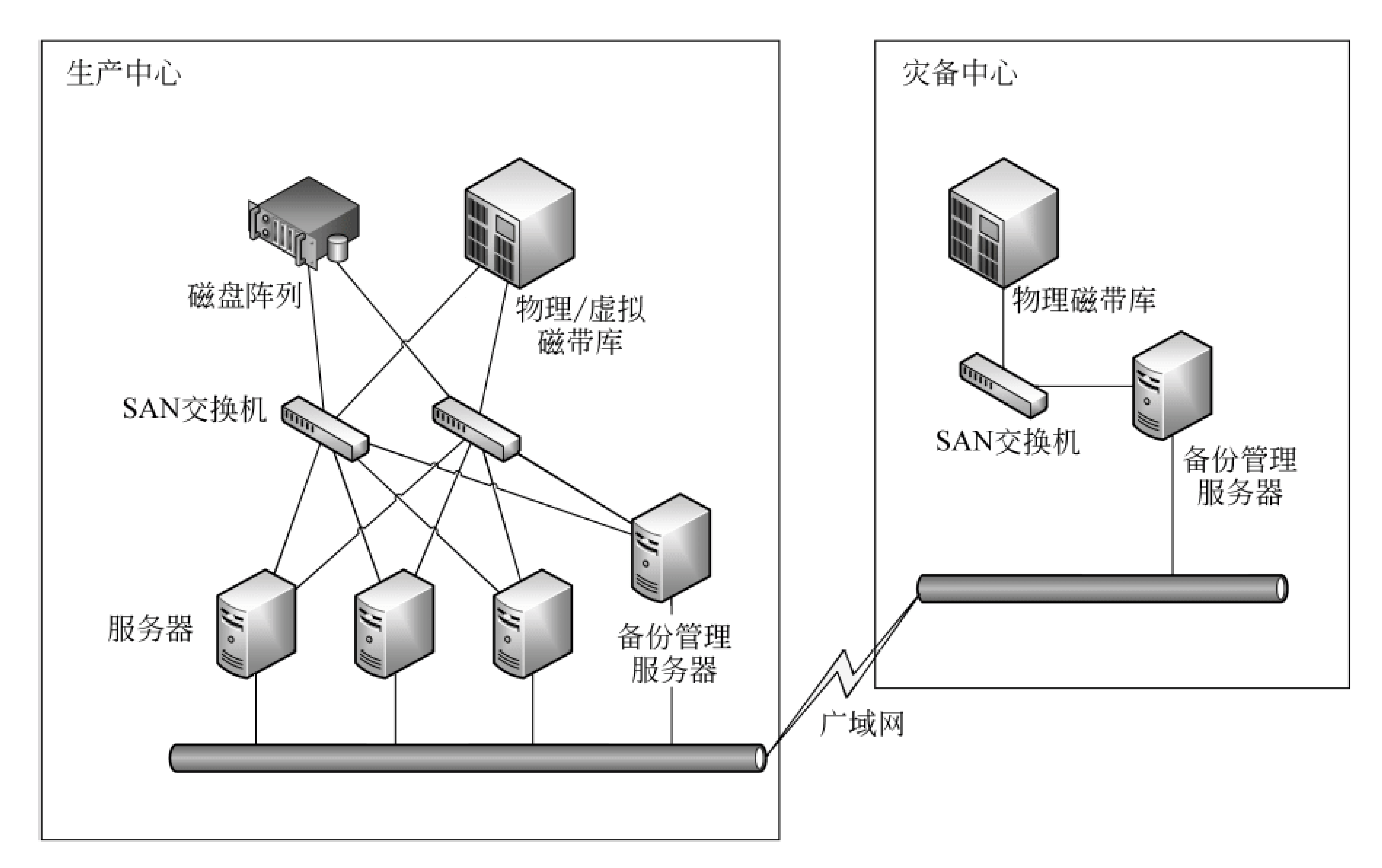


图 6-3　远程数据备份方案示意图

备份服务器：一是通过 FC 端口连接到光纤交换机实现与备份服务器进行连接；二是通过 iSCSI 端口与备份服务器通过 IP 端口进行连接。物理带库则一般通过 FC 或 SCSI 端口与备份服务器进行连接。

③ 在灾备中心配置物理带库和备份服务器，物理带库通过 FC 端口与备份服务器进行连接，同时在灾备中心部署备份软件，灾备中心的备份服务器将按照生产中心备份管理服务器的作业调度策略将生产数据备份到灾备中心的物理带库上。

(3) 方案特点

本方案是通过两个备份流实现生产中心与灾备中心的远程备份：一是本地被分流，即将生产数据通过本地备份软件备份到生产中心的物理带库或虚拟带库上；二是将生产中心的物理带库或虚拟带库上的数据通过备份策略备份到灾备中心的物理带库上。

按照本方案，生产中心可采用虚拟带库，也可采用物理带库作为备份设备，而灾备中心采用物理带库作为备份设备；由于虚拟带库具有磁盘读取快、备份效率高、恢复时间短等特点，因此采用虚拟带库时，可提高数据本地备份和恢复的速度。但虚拟带库是将硬盘作为数据备份的物理介质，因此不具备数据的可移动长期保存的特点，而物理带库是将磁带作为数据备份的物理介质，磁带备份后可存放到单独的空间进行长期保存，因此各地可根据自身的实际情况选择合适的备份设备。

(4) 适用范围

本方案适用于生产数据较少且备份网络带宽较高的信息系统备份。

(5) 可选备份软件

根据本方案的备份要求，可选用 Symantec Netbackup、IBM TSM、Bakbone NetVault Backup、EMC Networker 等备份软件实现备份。

6.2　基于目的端重复数据删除技术的备份方案介绍

6.2.1　技术描述

该方案是利用虚拟带库本身具有的重复数据删除技术识别备份到虚拟带库上的归档备份数据文件的冗余数据段，并进行删除，以减少数据备份量，提高备份效率。生产中心与灾备中心之间的数据备份是通过两端虚拟带库之间的数据备份实现的，生产中心的虚拟带库将冗余数据段删除后通过 IP 网络传输到灾备中心的虚拟带库上，如图 6-4 所示。

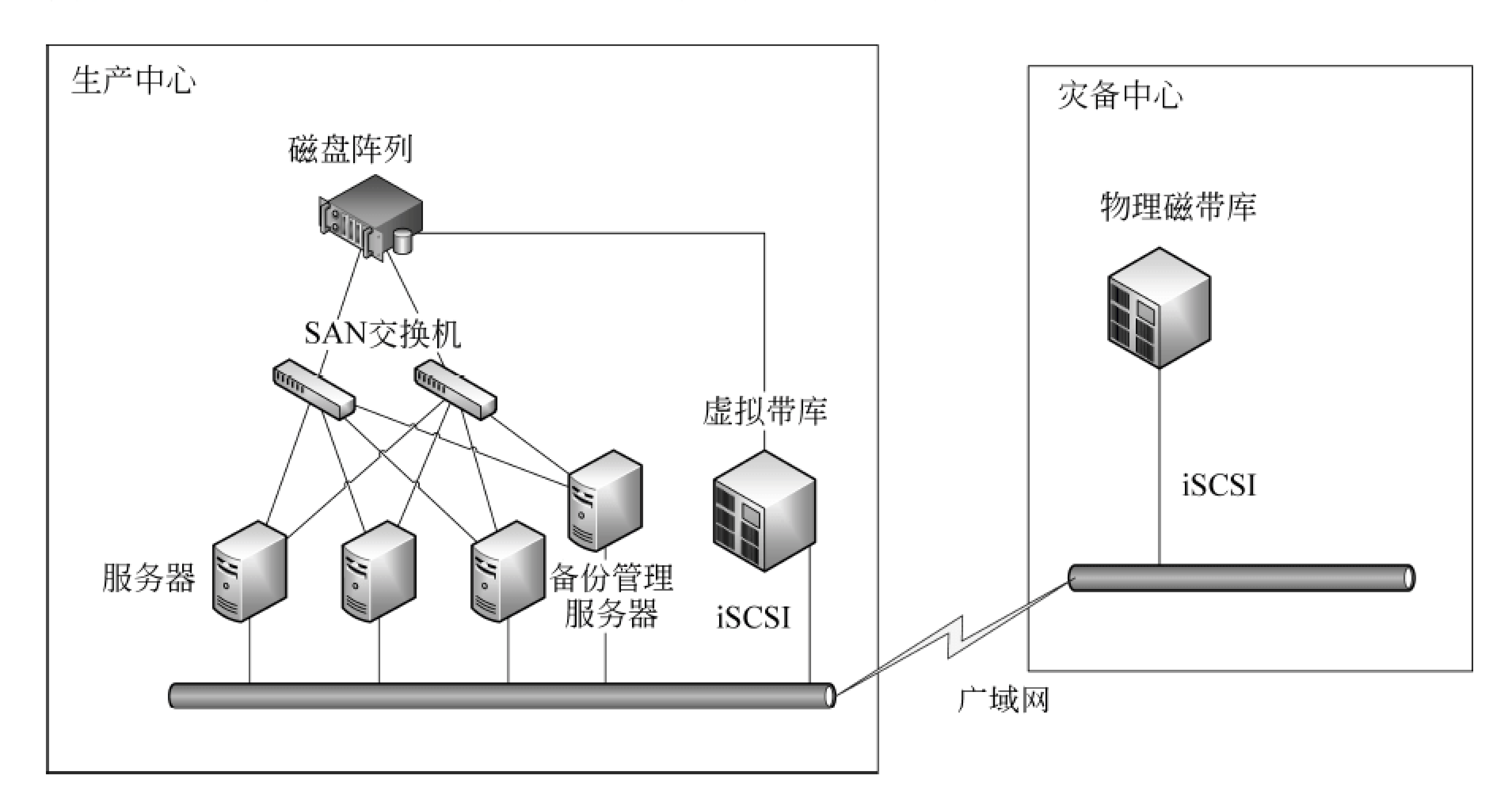


图 6-4　基于虚拟带库数据备份方案示意图(1)

如果灾备中心有备份介质保存的需求，则可在灾备中心再部署一台物理带库，灾备中心虚拟带库将备份的数据通过离线的方式备份到物理带库上，如图 6-5 所示。

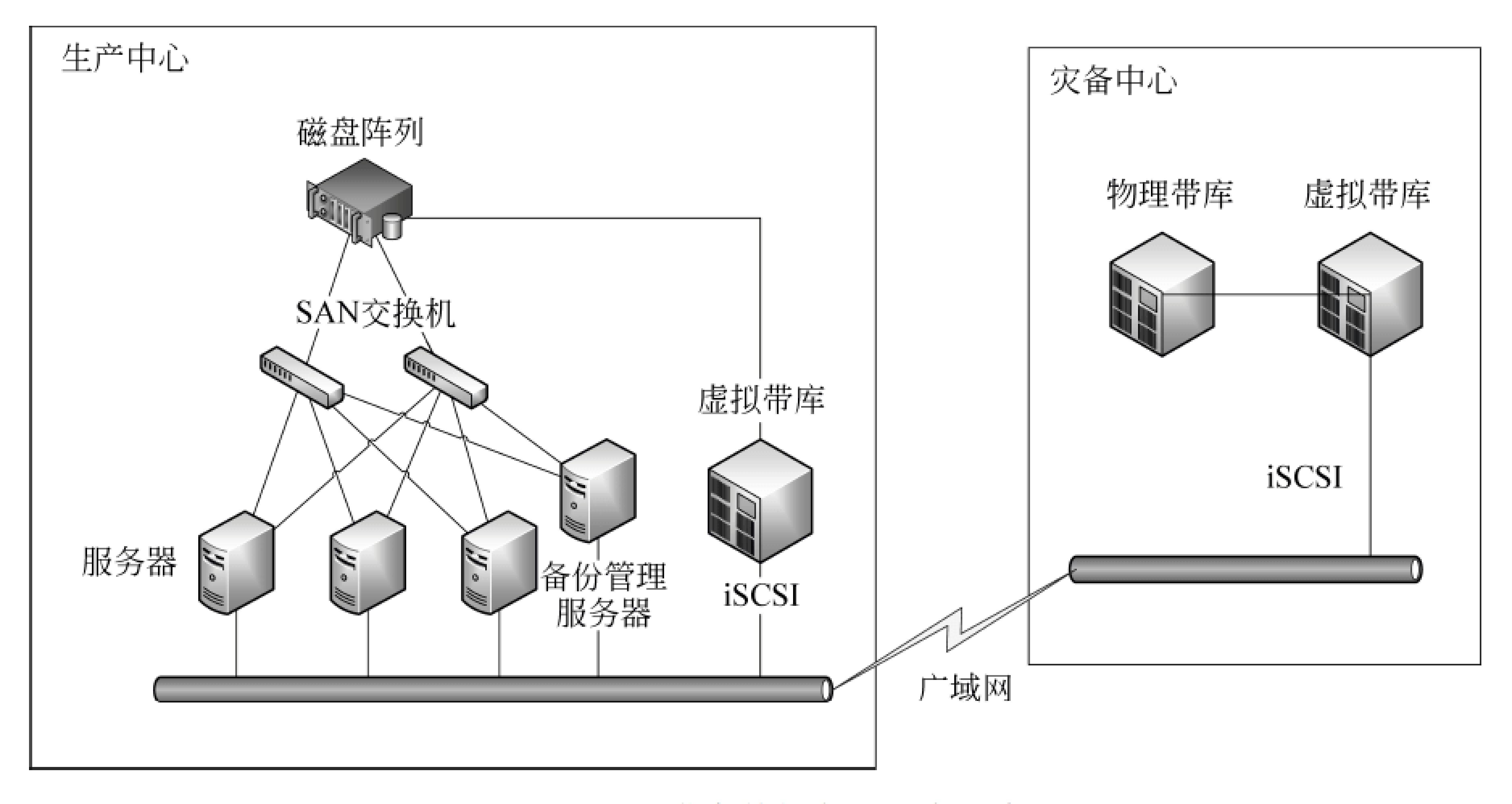


图 6-5　基于虚拟带库数据备份方案示意图(2)

6.2.2 资源配置要求

(1) 在生产中心配置备份服务器,并在备份服务器上部署备份软件,备份服务器通过HBA卡与光纤交换机进行连接,以保证备份服务器能通过SAN网访问到生产数据的存储设备。

(2) 在生产中心配置虚拟带库,虚拟带库可通过两种方式连接备份服务器:一是通过FC端口连接到光纤交换机实现与备份服务器进行连接;二是通过iSCSI端口与备份服务器通过IP端口进行连接。

(3) 在灾备中心配置虚拟带库,虚拟带库通过iSCSI端口与本地的IP交换机进行连接;如果要实现异地介质备份,可在灾备中心配置物理带库,通过离线备份的方式实现灾备中心虚拟带库与物理带库之间的离线数据备份。

6.2.3 技术特点

(1) 整个备份过程由有三个备份流组成的:一是生产中心生产的数据到虚拟带库的本地备份;二是生产中心与灾备中心虚拟带库之间通过IP网络的远程备份;三是灾备中心虚拟带库到物理带库的离线备份(如果有异地介质存放);由于备份数据在传输前已删除了冗余数据段,因此减少了数据复制量(数据量可减少至1/20～1/50),降低网络带宽资源和备份设备的能耗。

(2) 生产中心和灾备中心的虚拟带库之间可通过加密方式进行传输,实现数据备份的安全性。

(3) 可实现多点间的数据复制。

6.2.4 适用范围

本方案适用于地市中心直接将数据备份到省中心,再由省中心备份到本地磁带上。

6.2.5 可选技术

(1) 本地数据备份软件可选择 Symantec Netbackup、Bakbone NetVault、IBM TSM、EMC Networker。

(2) 虚拟带库及重复数据删除技术可选择 EMC EDL/DataDomain、昆腾 DXi 系列、SUN STK Sepaton。

6.3 基于源端重复数据删除技术的备份方案介绍

6.3.1 技术描述

基于源端重复数据删除技术的备份方案主要是删除源生产数据文件的冗余数据段。其主要实现方式是在生产中心将生产原数据文件传输到本地的备份存储池内,存储池将数据文件拆分成小的数据段,通过段级重复数据删除后,备份管理软件只将包含新数据或修改数据的部分备份到灾备中心的备份存储池中,备份到灾备中心后的文件主要部分保持不变,不

需要重新备份。从而实现了段级别增量备份。除此以外，该技术还可以实现每日全备份恢复以减少恢复时间，如图 6-6 所示。

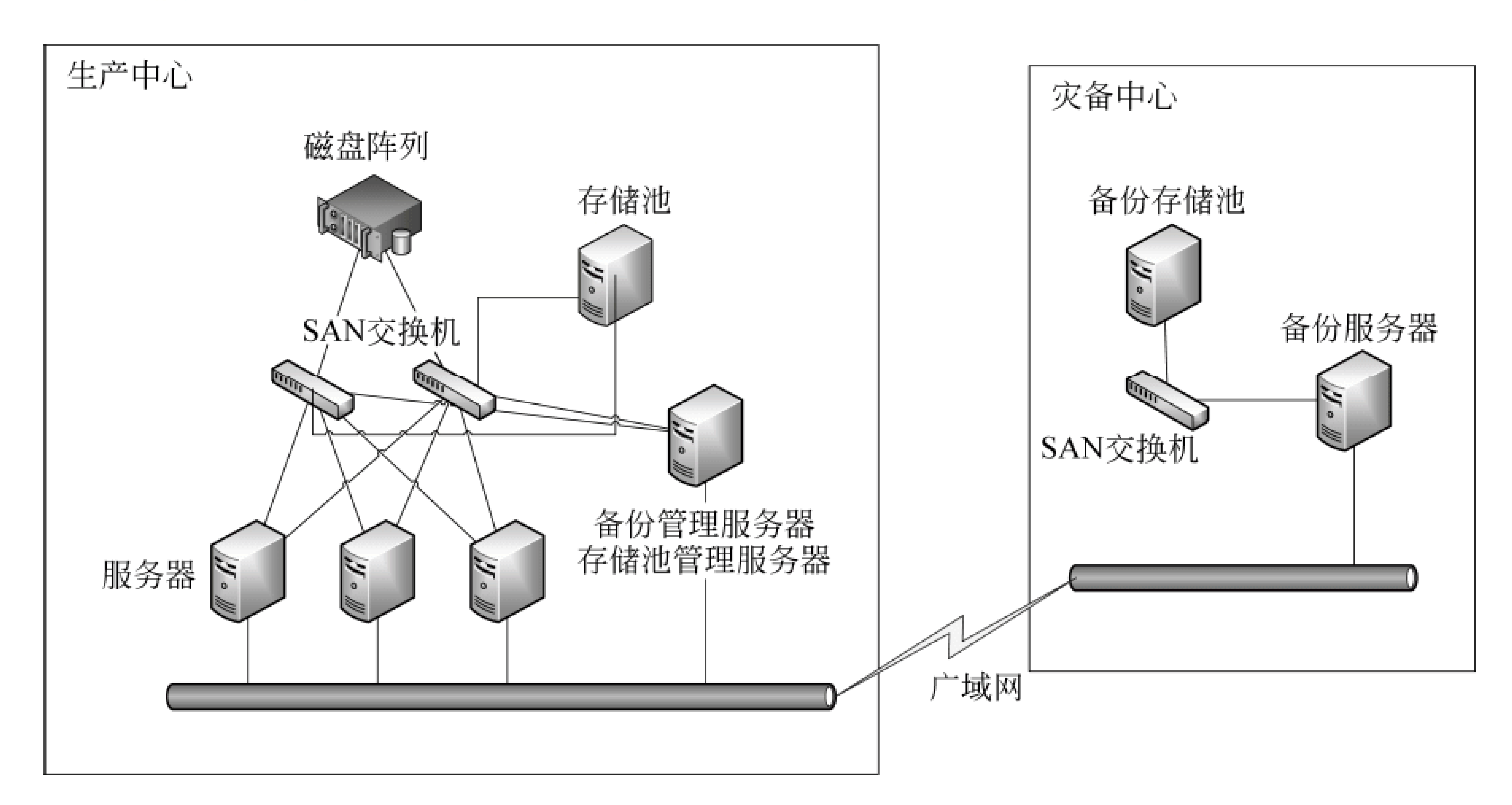


图 6-6　基于源端重复数据删除技术的备份示意图

为节省设备投资，并达到备份效果，生产中心的存储池和灾备中心的备份存储池可采用低端磁盘阵列。

采用这种技术进行数据备份时，生产中心的备份管理服务器可采用自动备份和手动备份两种方式，灾备中心备份存储池中的数据还可由备份服务器定时归档迁移至磁带库，并保存在磁带上，以保证灾备中心具有可用于长期保存的归档数据。

6.3.2　资源配置要求

(1) 在生产中心配置管理服务器，用于备份管理和存储池管理，管理服务器通过 HBA 卡与光纤交换机进行连接，以保证其能通过 SAN 网访问到生产数据的存储设备，并能将数据通过 SAN 网存放在存储池上。

(2) 在生产中心可配置低端磁盘阵列作为存储池，存储池通过 FC 或 IP 端口与备份服务器进行连接。

(3) 灾备中心配置管理设备，该设备按照既定调度策略接收备份数据，并将其存放在备份存储池中。

(4) 在灾备中心可配置低端磁盘阵列作为备份存储池，用于存放消重后的备份数据，备份存储池通过 FC 或 IP 端口与管理设备进行连接。

(5) 为实现异地介质备份，可在灾备中心配置物理带库，通过备份服务器将备份存储池中的备份数据归档到磁带上。

提示： 对于生产中心和灾备中心的管理服务器和存储池，可采用一台专用的存储设备，该设备同时具有备份管理、存储池管理和存储池的功能。

6.3.3　技术特点

(1) 由于生产数据在传输前已进行了冗余数据段删除，因此减少了数据复制量(数据量

可减少至1/500),降低网络带宽资源(网络带宽可节省至1/50)和备份设备的能耗。

(2) 当数据恢复时,只需将灾备中心的消重数据直接进行恢复,从而减少恢复次数。

(3) 提供备份数据加密功能,确保备份数据的安全。

(4) 通过存储池管理和完善的报表功能,将生产中心的存储池和灾备中心的备份存储池统一管理起来,降低了设备和介质管理的复杂度。

6.3.4 适用范围

本方案适用于地市中心与省中心网络带宽较低的情况。

6.3.5 可选技术

重复数据删除技术可选择 EMC Avamar、Symantec Netbackup PureDisk。

6.4 基于智能存储设备的数据复制技术

6.4.1 技术描述

基于智能存储设备的数据复制技术采用先进的智能存储复制软件,通过光纤直连、SDH、ATM 或 IP 网络等在灾备中心与生产中心建立磁盘镜像连接,实现数据的 7×24 小时远程实时复制。

智能存储数据复制采用基于存储控制器的复制技术,通过存储系统微码提供的数据复制功能,将源磁盘数据复制至目标磁盘。智能存储数据复制技术与主机平台无关,可实现异构平台环境下的数据远程备份。

远程存储数据复制技术按请求复制的主机是否需要远程镜像站点的确认信息,又可分为同步远程复制和异步远程复制。

同步复制技术是指通过远程复制软件,将本地数据以完全同步的方式复制到异地,每一本地的 I/O 事务均需等待远程复制的完成确认信息,才予以释放。同步镜像使远程拷贝总能与本地机要求复制的内容相匹配。当主站点出现故障时,用户的应用程序切换到备份的替代站点后,被复制的远程副本可以保证业务继续执行而没有数据的丢失。但它存在往返传播造成延时较长的缺点,只限于在相对较近的距离上应用。

异步复制技术保证在更新远程存储视图前完成向本地存储系统的基本 I/O 操作,而由本地存储系统提供给请求镜像主机的 I/O 操作完成确认信息。远程的数据复制是以后台同步的方式进行的,这使本地系统性能受到的影响很小,传输距离长(可达1000千米以上),对网络带宽要求小。但是,许多远程的从属存储子系统的写没有得到确认,当某种因素造成数据传输失败,可能出现数据一致性问题。为了解决这个问题,目前大多采用延迟复制的技术(本地数据复制均在后台日志区进行),即在确保本地数据完好无损后进行远程数据更新。

6.4.2 技术架构及特点

基于智能存储的数据复制技术具有高效快速的优点,能较好地保证数据的完整性和一致性,数据的复制备份过程不占用主机资源,操作控制比较简单。但由于基于存储的数据复

制技术对于主、备份中心之间的网络条件(稳定性、带宽、链路空间距离)要求较苛刻。对于带宽情况不理想的远距离数据传输需要较长的时间。较高的网络带宽要求尤其是长途线路会大大增加日常运营成本。同时基于智能存储的复制技术开放性较差,不同厂家的存储设备系统一般不能配合使用,不利于投资保护。

针对全省大集中的灾备体系架构,该技术也有对应的三种架构,即 A—B 模式、A—B—C 模式和 A—B/A—C 模式,其逻辑架构如下:

6.4.3　生产—同城复制模式(即 A—B 模式)

生产中心与同城灾备中心采用基于磁盘阵列的同步复制模式,这种复制模式在向远程镜像磁盘卷中写入数据时,复制过程如下。

(1) 接收生产中心主机的写 I/O 操作。

(2) 将数据写到生产中心本地磁盘阵列的缓存中。

(3) 通过链路将其送到同城灾备中心磁盘阵列的缓存中。

(4) 同城磁盘阵列向生产磁盘阵列发送数据接收确认信号。

(5) 修改磁道表。

(6) 通知生产主机 I/O 操作完成。

如图 6-7 所示,这种模式需要跟踪生产中心每一个磁盘卷的变化情况,并将变化量同步复制到同城灾备中心,因此对网络带宽的要求比较高。

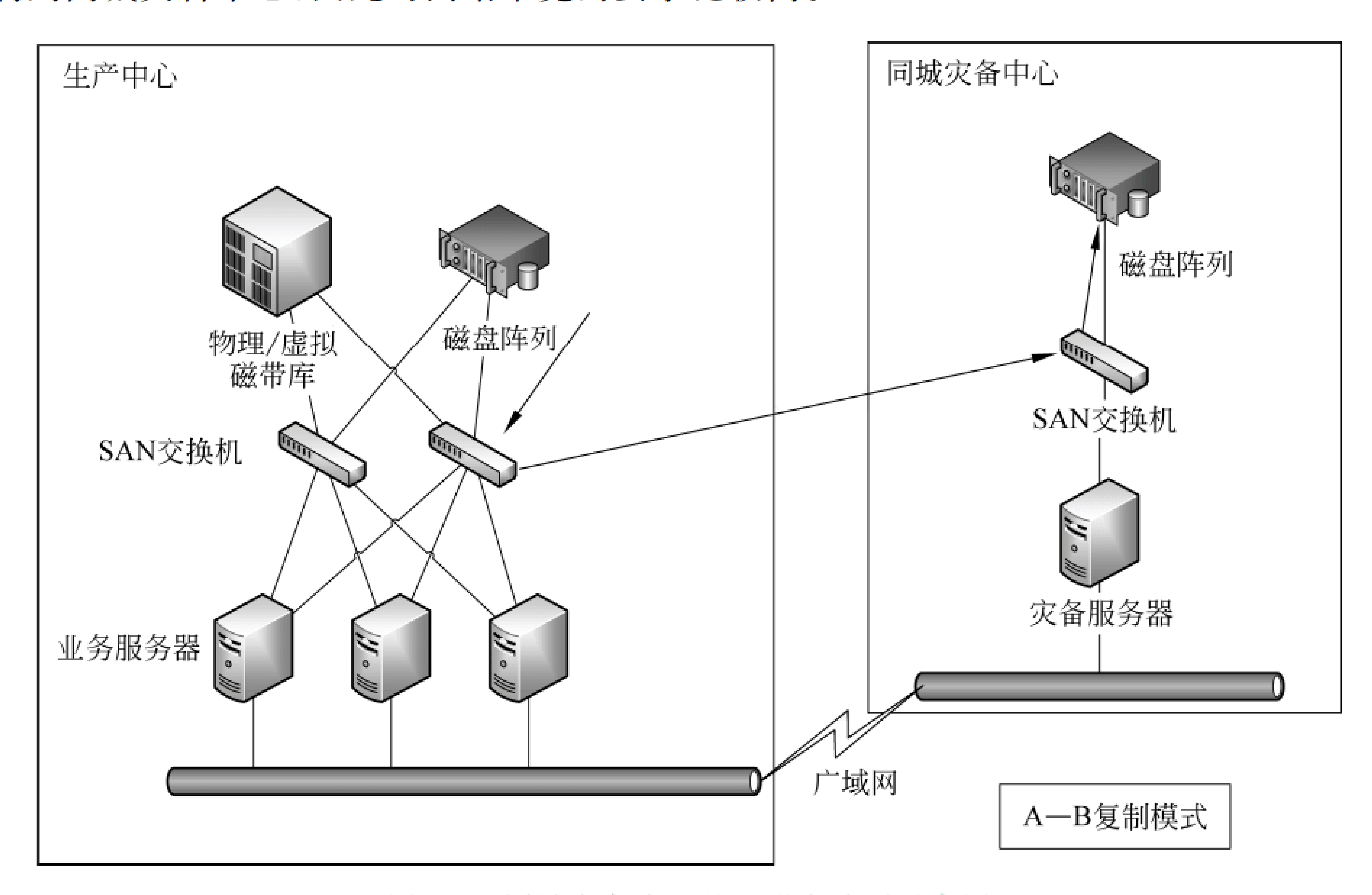


图 6-7　同城灾备中心基于磁盘阵列示意图

1. 生产—同城及生产—异地(即 A—B 及 A—C 模式)

生产中心将数据同步复制到同城灾备中心的同时,通过广域网专线将数据以异步的方式复制到异地灾备中心,其复制过程如下。

(1) 接收生产中心主机写 I/O。

（2）将数据写到生产中心本地磁盘阵列的缓存中。

（3）向生产主机发回设备结束信号。

（4）通过链路将其送到同城灾备中心磁盘阵列的缓存中。

（5）同城磁盘阵列向生产磁盘阵列发送数据接收确认信号修改磁道表。

如图 6-8 所示，异步复制的网络带宽要求取决于两端磁盘阵列缓存的大小、RPO 时间要求和数据变化量等因素，应根据实际环境的测量来确定线路带宽，但由于业务的实时性要求，随时都有可能产生变化的磁盘卷，因此也需要进行专线连接，为避免数据复制的单点故障，建议在同城灾备中心和异地灾备中心也申请一条专线作为生产中心到异地灾备中心的备份复制线路，保证中心复制的高可用性。

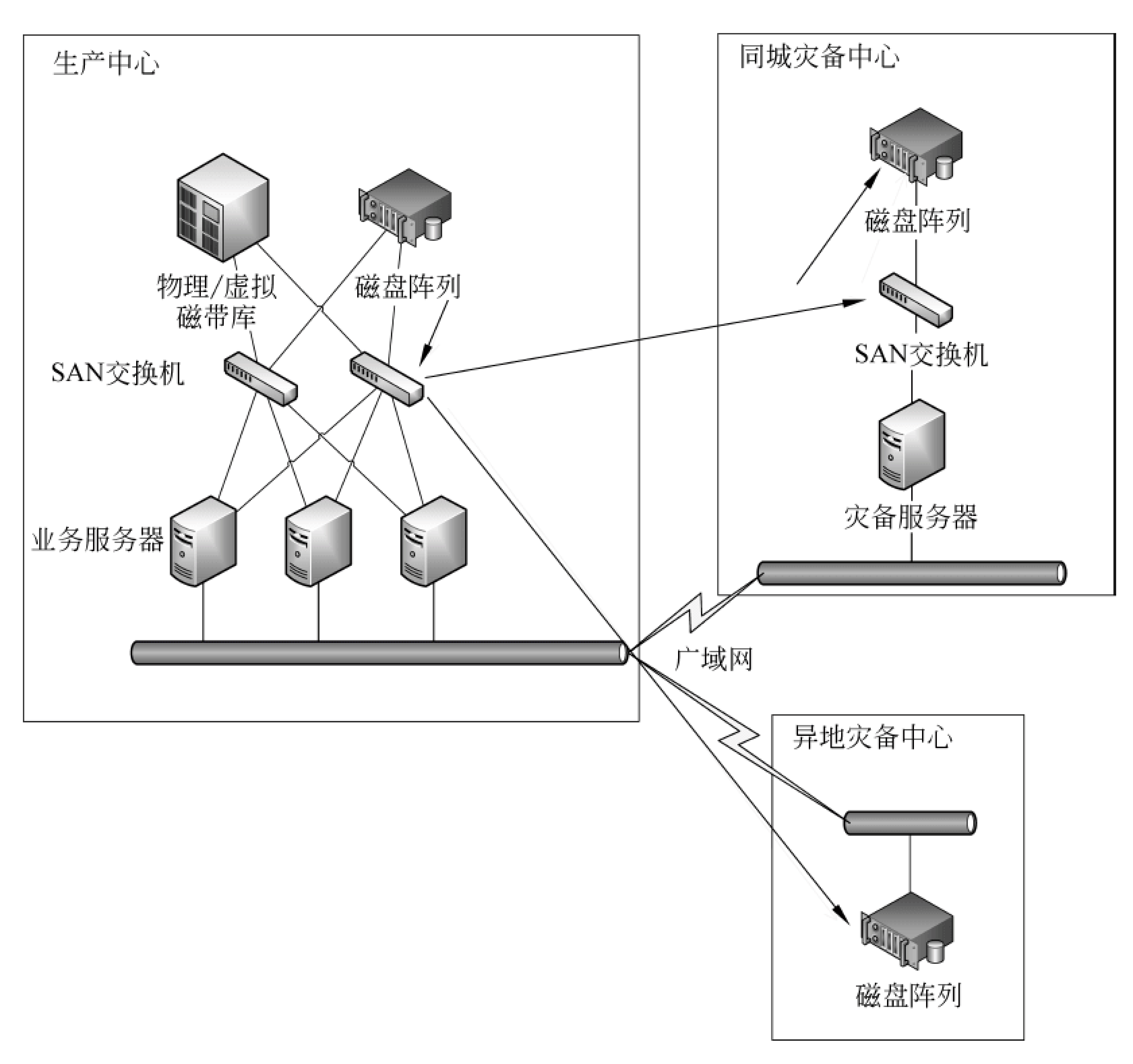


图 6-8　两地三中心灾备示意图（A—B、A—C 复制模式）

2. 生产—同城—异地（即 A—B—C 模式）

生产中心将数据同步复制到同城灾备中心后，同城灾备中心通过广域网专线将数据以异步的方式复制到异地灾备中心，复制过程同上。这种方式的数据复制不是由生产中心发起，而是由同城灾备中心发起，因此这种复制技术的好处是异步复制对生产中心业务运行的影响比较小。

如图 6-9 所示，在这种模式下，同城与异地灾备中心之间申请一条专线用于异步复制，将同城灾备中心的数据异步复制到异地灾备中心。

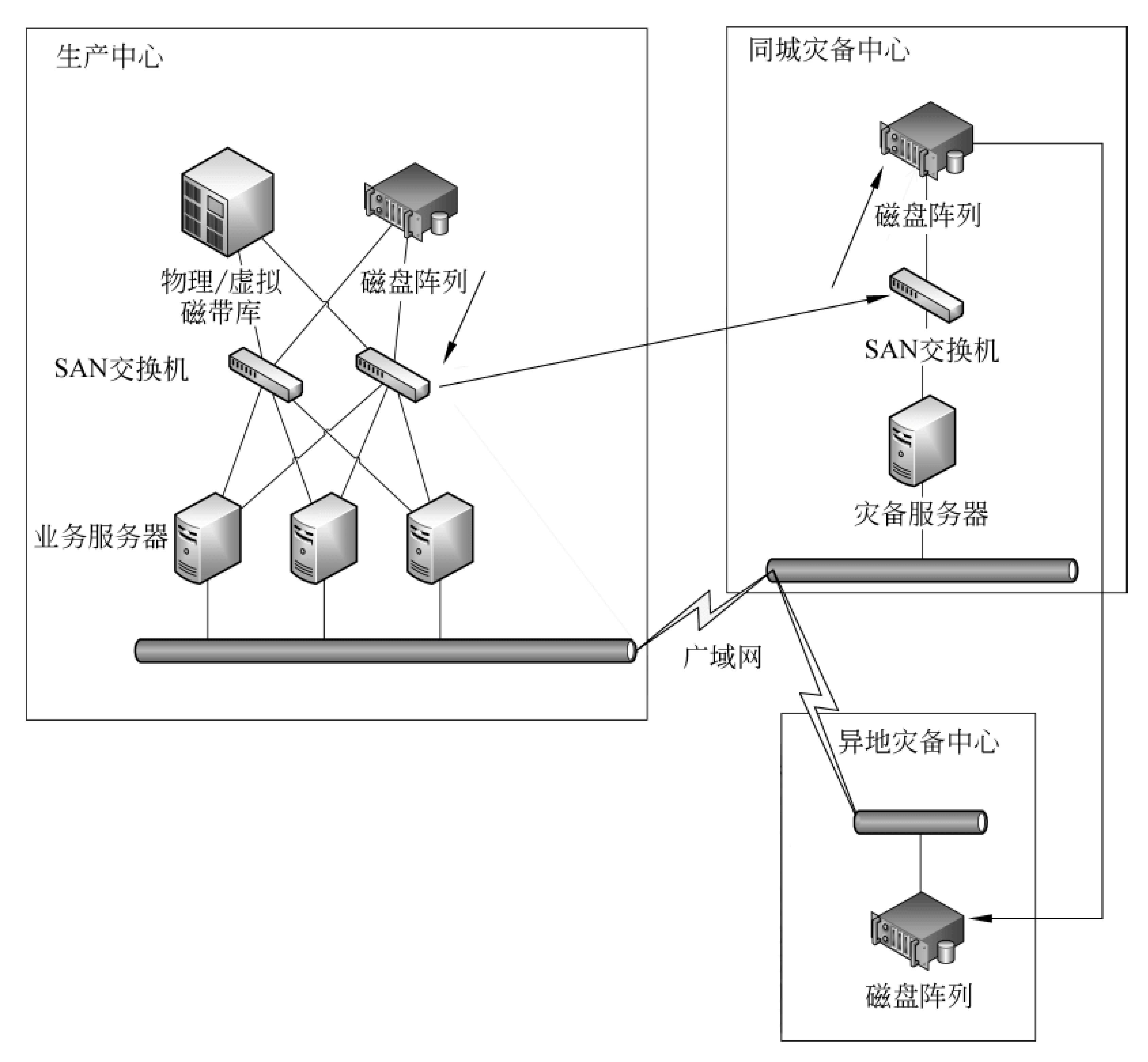


图 6-9　两地三中心灾备示意图(A—B—C 复制模式)

6.4.4　资源配置要求

生产中心和灾备中心的资源配置要求如下。

(1) 生产中心和灾备中心各配置同构并且支持相同的远程数据复制的存储设备。

(2) 在生产中心和灾备中心分别部署远程存储复制软件。

(3) 由于该技术与主机无关,因此数据复制通过 SAN 网络实现,同步一般要求采用光纤链路,异步一般采用专线连接,同步比异步要求的带宽高。

(4) 网络线路选择方式如下。

① 生产—同城复制模式适用于地市或省生产中心通过光纤或专线连接同城灾备中心,如果采用同步复制技术,应采用光纤连接;如果采用异步复制技术,应采用专线连接;对于同城数据级灾备模式,则生产中心可采用一条物理线路(裸光纤或专线)连接灾备中心,用于生产中心与同城灾备中心之间的数据复制;对于同城应用级灾备模式,则生产中心与同城灾备中心应采用裸光纤+波分复用设备进行连接。

② 对于异地数据级灾备模式,由于异地灾备中心与生产中心相距较远,租用光纤费用太高,因此一般采用专线实现生产中心或同城灾备中心与异地灾备中心之间的异步数据复制。

③ 网络带宽计算

最小带宽=平均数据变化量÷RPO÷系数

其中系数=有效带宽÷线路带宽,一般为 0.3～0.4。

6.4.5 适用范围

(1) 对于地市集中情况,可采用同城数据级或应用级灾备模式。

(2) 对于省中心,可采用同城数据级或应用级灾备模式,以及异地数据级灾备模式。

6.4.6 主流技术

基于智能存储设备复制技术主要包括 EMC SRDF/MirrorView、HDS TrueCopy/HUR、IBM PPRC、NetApp SnapMirror、HP Continuous Access 等。

6.4.7 实施步骤

基于存储的数据复制技术的灾备系统较复杂,其实施过程周期较长,实施的部分环节需要生产中心的配合,对生产中心的运营维护有一定影响,应避开业务繁忙期进行项目实施作业。项目实施的大致步骤如下。

(1) 根据选用的存储设备数据复制技术和生产系统对数据的时间点要求,进行前期论证。

(2) 在生产中心智能存储设备上采集实际生产状态数据,作为存储规划基本参数。

(3) 根据设定数据量和复制操作时间窗口对数据复制需要的电子线路类型和带宽进行规划。

(4) 采购与所选数据复制技术兼容的存储设备,根据生产系统要求进行设备初始化及全量数据同步。

(5) 生产与灾备两端存储根据数据复制技术进行配置、策略设置等操作。

(6) 测试存储设备间数据复制策略是否达到设计要求。

(7) 测试生产系统数据存储设备与灾备数据存储设备的切换。

(8) 测试灾备应用系统与灾备数据存储设备的读写等应用系统功能。

(9) 通过测试后,投入试运行。

(10) 试运行正常后转入正式运行。

6.5 基于数据库的数据复制技术

6.5.1 技术描述

基于数据库的数据复制技术利用的是数据库系统所提供的日志备份和恢复机制,在生产中心正常工作的同时产生归档日志文件(Archived Log)或重做日志不断地传送到灾备中心,并且利用这些日志文件在灾备中心上连续进行恢复(Recover)操作,以保持灾备系统与生产系统的一致。当生产中心发生故障时,使用备份的日志文件在灾备中心恢复生产中心的数据。

通过使用基于数据库软件复制技术,保证远程数据库的复制。生产中心主机安装数据库同步软件的客户端和数据库代理,通过搭建的网络环境与灾备中心数据库同步软件的服

务器端通信,按照定义的规则实现整库级、用户级、表级、日志级的数据同步。在生产中心的生产服务器上部署数据库同步软件的客户端和数据库代理,可以和灾备中心的数据库同步软件服务器端实现 1 对多模式的远程数据复制,如图 6-10 所示。

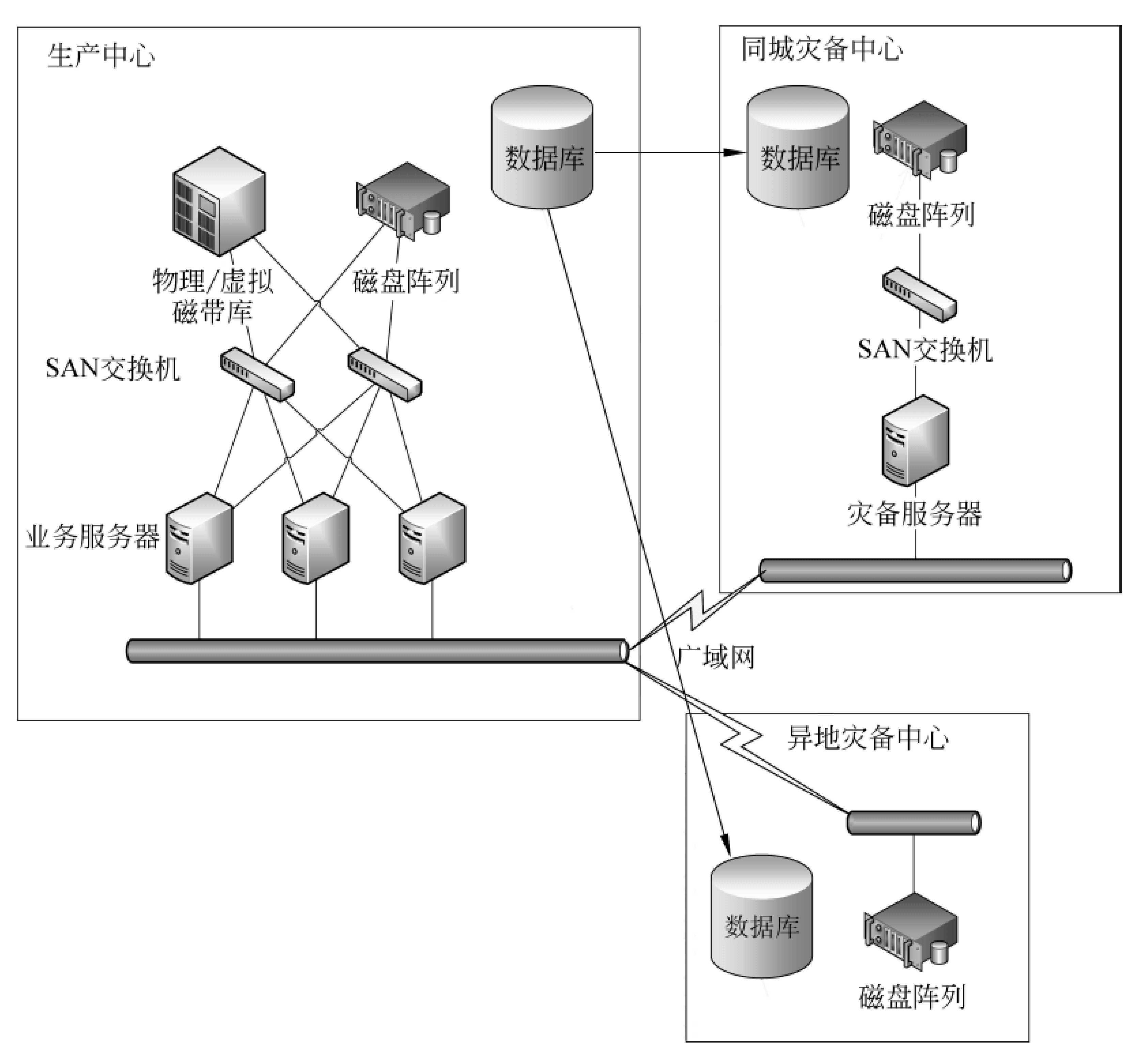


图 6-10　基于数据库的两地三中心示意图(数据库复制模式)

6.5.2　资源配置要求

(1) 灾备中心可配置与生产中心异构的存储设备,但其性能和容量应与生产中心相当。

(2) 在生产中心和灾备中心部署基于数据库的复制软件。

(3) 生产中心和灾备中心通过 IP 网络(一般是专线)进行实现数据复制。

6.5.3　技术特点

基于数据库的数据复制技术具有高效快速的优点,能较好地保证数据的完整性和一致性,对于生产中心和灾备中心之间的网络条件(稳定性、带宽、链路空间距离)要求也较低。但数据的复制过程占用主机资源,对生产系统有一定影响,对于交易较频繁、生产数据库压力较大的应用系统,选择该方式时需要经过充分的论证。另外,该复制模式只是针对数据库数据,对于文件系统数据等其他类型的数据,则需要另行考虑其他数据复制方案,因此最终形成的综合方案较复杂。该技术对数据库部署及管理能力提出了较高的要求,为项目实施和日常运行维护带来了一定的不便。

6.5.4 主要实施步骤

基于数据库的数据复制技术的灾备系统，其实施过程周期较长，实施的部分环节需要生产中心的配合，对生产中心的运营维护有一定影响，应避开业务繁忙期进行项目实施作业。

项目实施的大致步骤如下。

（1）根据生产系统对数据的时间点要求，进行前期论证。

（2）根据日志数据量和复制操作时间窗口对数据复制需要的电子线路类型和带宽就行规划。

（3）生产与灾备两端主机及数据库根据数据复制技术进行配置、策略设置等操作。

（4）测试数据复制策略是否达到设计要求。

（5）测试生产系统数据库与灾备数据库的切换。

（6）通过测试后，投入试运行。

（7）试运行正常后转入正式运行。

6.5.5 适用范围

数据库复制技术一般适用于应用级灾备模式。

6.5.6 主流技术

基于数据库的数据复制技术，主流的技术有 Oracle Dataguard、Quest SharePlex、GoldenGate、DSG RealSync、Sybase Replication Server 等多种可选技术。

6.6 基于主机的数据复制方案介绍

在这类数据复制方案中，主要有两种方案：一是数据卷镜像方案；二是数据卷复制方案。

6.6.1 数据卷镜像方案

1. 技术描述

数据卷镜像方案属于基于主机的复制技术的一种，这种方式的原理是在生产中心和灾备中心之间有光纤链路，并且在生产中心的所有需要复制的服务器上部署专业存储管理软件，在灾备中心部署相应的存储系统和主机，利用专业存储管理软件将生产中心存储系统和灾备中心的存储系统组成一个镜像存储系统，在生产中心的主机发生一个写操作时，利用专业存储管理软件的镜像功能，将这个写操作通过光纤链路同时传输至灾备中心的存储系统，在两个中心的存储系统都完成写操作时后，此操作才真正完成，这种数据卷镜像方案可以保证生产中心数据“零丢失”（RPO＝0），如图 6-11 所示。

由于采用了生产中心和灾备中心存储系统镜像的模式，因此在两中心任何一方的存储系统出现故障（或性能低下）时，为了不影响生产中心的业务系统的正常运转，专业存储软件会将故障存储（或性能低下的存储）自动剔除出镜像系统，而由正常一方的存储系统独自承

担业务，这种由一方存储系统故障造成的灾难，其灾难接管工作是由专业存储管理软件自动进行的，无须人工干涉，即无停机时间(RTO=0)，如图 6-12 所示。

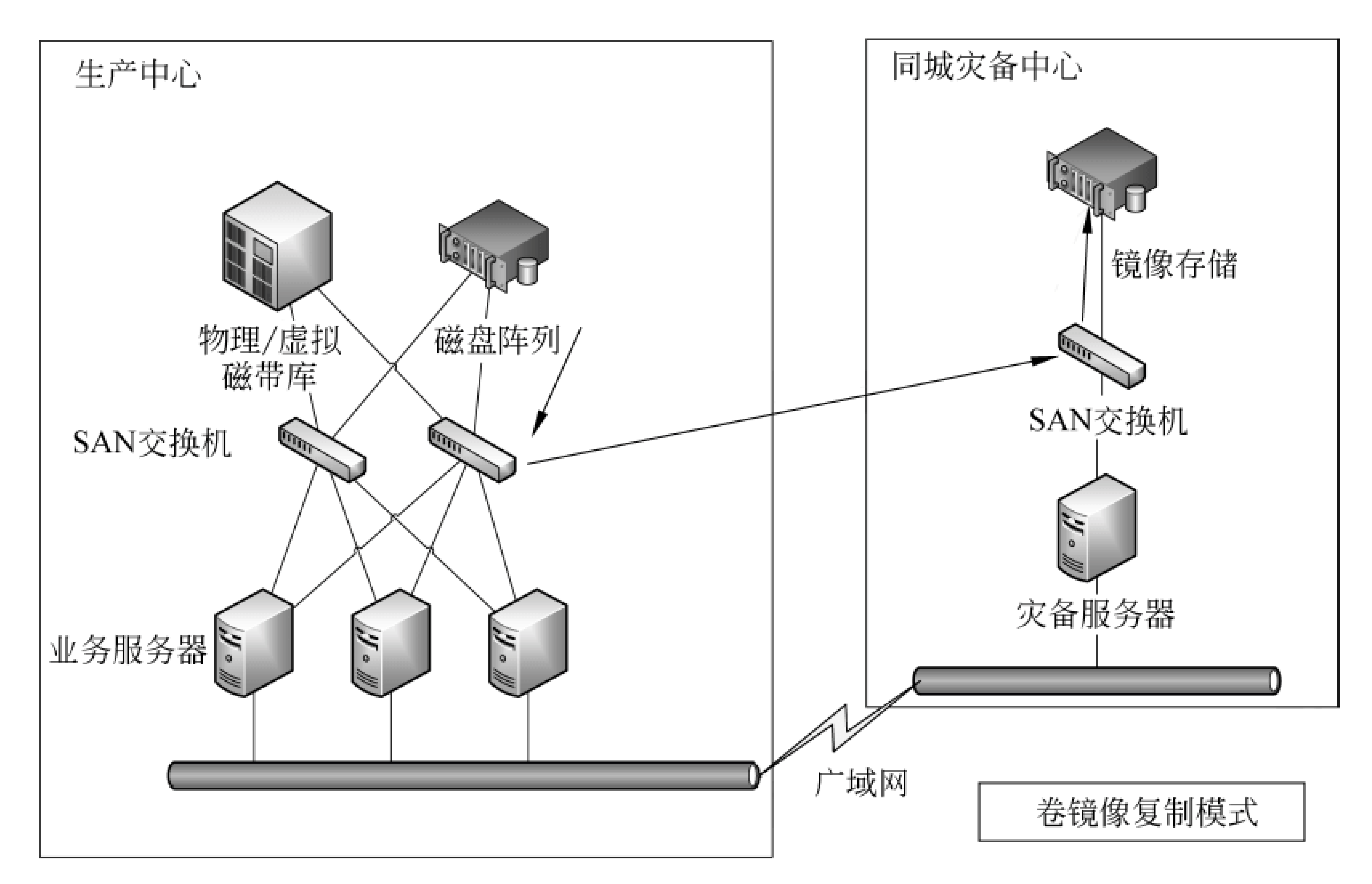


图 6-11　基于数据卷镜像示意图(1)

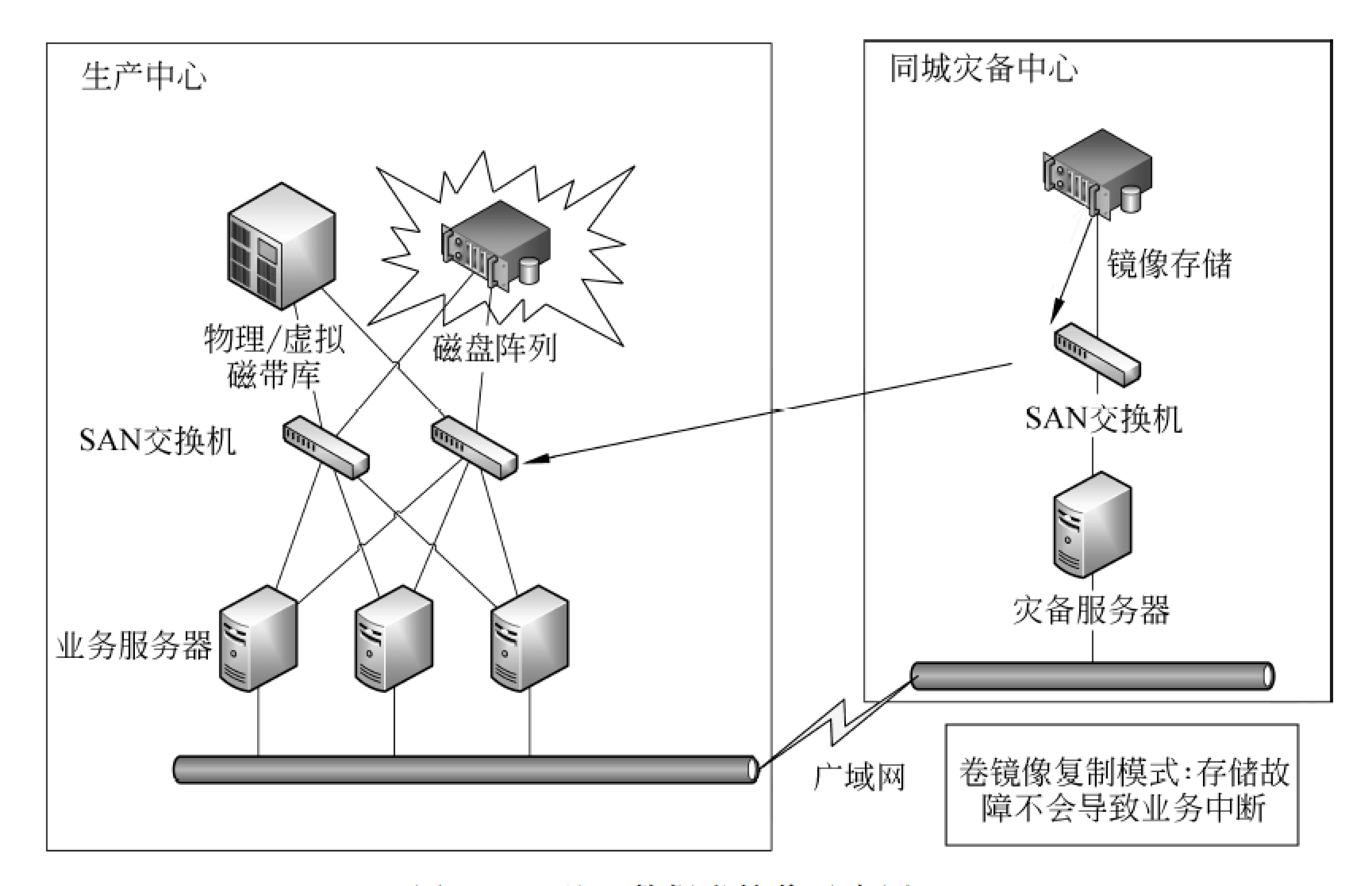


图 6-12　基于数据卷镜像示意图(2)

在生产中心的主机发生故障时，业务会发生中断，此时如需将应用切换至灾备中心，则可以采用集群管理软件的灾难切换功能，将生产中心的应用在很短的时间内自动切换至灾备中心的主机上，以保证业务的连续性，如图 6-13 所示。

2. 资源配置要求

(1) 生产中心和灾备中心需具备光纤通路。

(2) 灾备中心存储系统与生产中心存储系统性能相近，但无须同品牌。

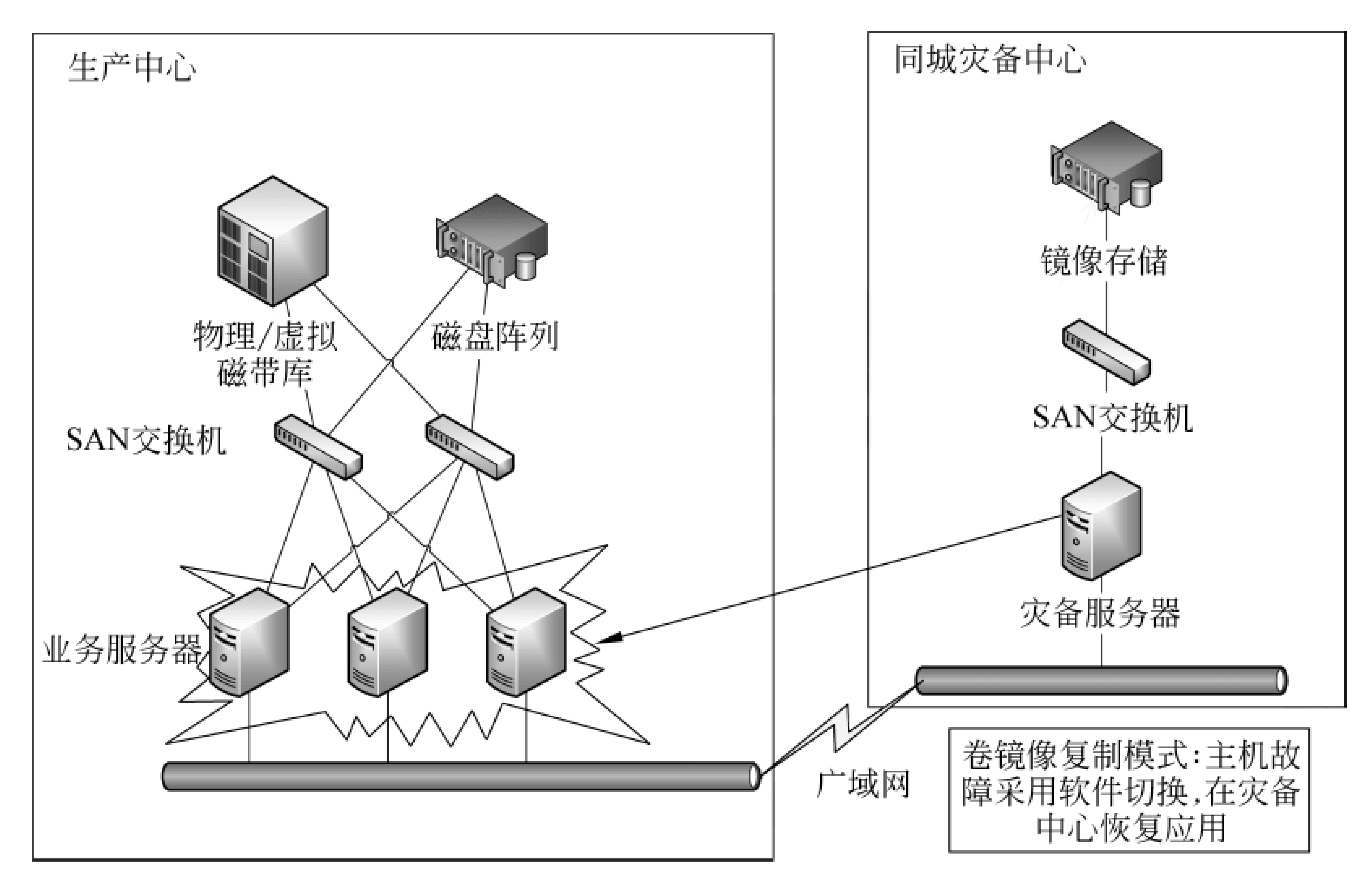


图 6-13　基于数据卷镜像示意图(3)

(3) 两中心距离不超过 100 公里。

(4) 在生产中心相应的服务器上需要配置 Symantec Storage Foundation 软件。

(5) 如果需要灾难自动切换功能,则需要在生产中心和灾备中心的相应服务器上均配置 Symantec Storage Foundation HA/DR 软件。

(6) 网络带宽需求=平均数据增量×(峰值持续时间-RPO)/(峰值数据增量×峰值持续时间-RPO)。

3. 适用范围

数据卷镜像灾备解决方案是生产-同城复制模式(即 A-B 模式),适用于对 RPO 及 RTO 要求较高,实时性要求较强的系统(如医保系统),针对这些系统,数据卷镜像灾备解决方案可以提供“零数据丢失”和“一键式切换”解决方案以保证其业务的连续性。

4. 技术特点

(1) 优点

① 零数据丢失:采用数据卷镜像灾备解决方案可以实现生产中心数据与灾备中心数据完全一致,即不存在数据丢失的可能性,即 RPO=0。

② 存储系统故障自动剔除:在生产中心或灾备中心任何一端存储系统出现故障时,专业存储管理软件均自动剔除故障存储系统,不会造成业务中断,即 RTO=0。

③ 在配置集群软件的前提下,灾难切换时间短,切换操作简易:在主机发生故障或整个生产中心失效时,可以采用专业存储管理软件的“一键式切换”功能,将生产中心的业务迅速自动切换至灾备中心,以保证业务的连续性。

④ 硬件选择余地较大:与基于智能存储系统的灾备方案不同,数据卷镜像灾备方案中生产中心和灾备中心存储系统无须同品牌。

(2) 缺点

对生产中心和灾备中心的链路带宽和稳定性有一定要求,由于采用数据卷镜像方式,因此需要两中心数据链路是光纤通道链路。

5. 实施步骤

数据卷镜像灾备解决方案的实施相对比较简单，一般无须更改任何硬件环境，而只需要在生产中心和灾备中心安装相应的专业存储管理软件，并配置两中心的存储镜像系统及“一键式切换”系统即可。项目实施的大致步骤如下。

(1) 根据生产系统对数据的时间点要求，进行前期论证。

(2) 对生产系统进行全备份(建议)。

(3) 在生产中心和灾备中心的服务器上安装专业存储管理软件。

(4) 停止业务访问(有停机时间)。

(5) 自动转换数据卷格式。

(6) 启动业务系统(恢复业务运行)。

(7) 在线配置生产中心和灾备中心镜像存储系统，自动进行数据初始化同步。

(8) 在配置集群软件的情况下，可在线配置“一键式切换”系统。

(9) 利用专业存储管理软件的“仿真模拟演练”功能进行灾备演练(可在线做，无须中断务、也无须中断灾备系统)。

(10) 通过测试后，投入试运行。

(11) 试运行正常后转入正式运行。

6.6.2 数据卷复制方案

1. 技术描述

数据卷复制方案是另一种常见的基于主机复制技术，与数据卷镜像方式相同的是，数据卷复制解决方案同样是利用专业存储软件进行容灾数据复制的，而与其不同的是，数据卷复制解决方案采用的是 IP 网络而非光纤链路进行复制，其复制方式大多采用异步复制方式，而其灾备距离不受限制，可以实现超远距离的灾备。

同样需要在生产中心和灾备中心的服务器上安装专业存储管理软件，并配置对应的数据卷的复制关系，数据初始化完毕后，生产中心主机每接收到一个写操作，都会同时通过 IP 链路向灾备中心的主机传送相同的写操作，灾备中心的主机将会把这个写操作在灾备中心的存储系统上完成，如图 6-14 所示。

事实上，数据卷复制灾备解决方案也是一种很好的集中灾备解决方案，可以实现高达 32 个逻辑数据卷对一个逻辑数据卷复制的功能，即支持多数据中心向一个灾备中心容灾的功能。

无论是生产中心的主机还是存储器发生故障时，业务都会发生中断，此时如需将应用切换至灾备中心，则可以采用专业存储管理软件中的灾难切换功能，将生产中心的应用在很短的时间内自动切换至灾备中心的主机上，以保证业务的连续性，如图 6-15 所示。

2. 资源配置要求

(1) 生产中心和灾备中心只需具备 IP 通路。

(2) 灾备中心存储系统与生产中心存储系统无须同品牌，灾备中心存储系统档次可以低于生产中心存储系统。

(3) 灾备中心主机操作系统类型与生产中心主机操作系统类型一致。

(4) 在生产中心和灾备中心相应的服务器上配置卷管理软件。

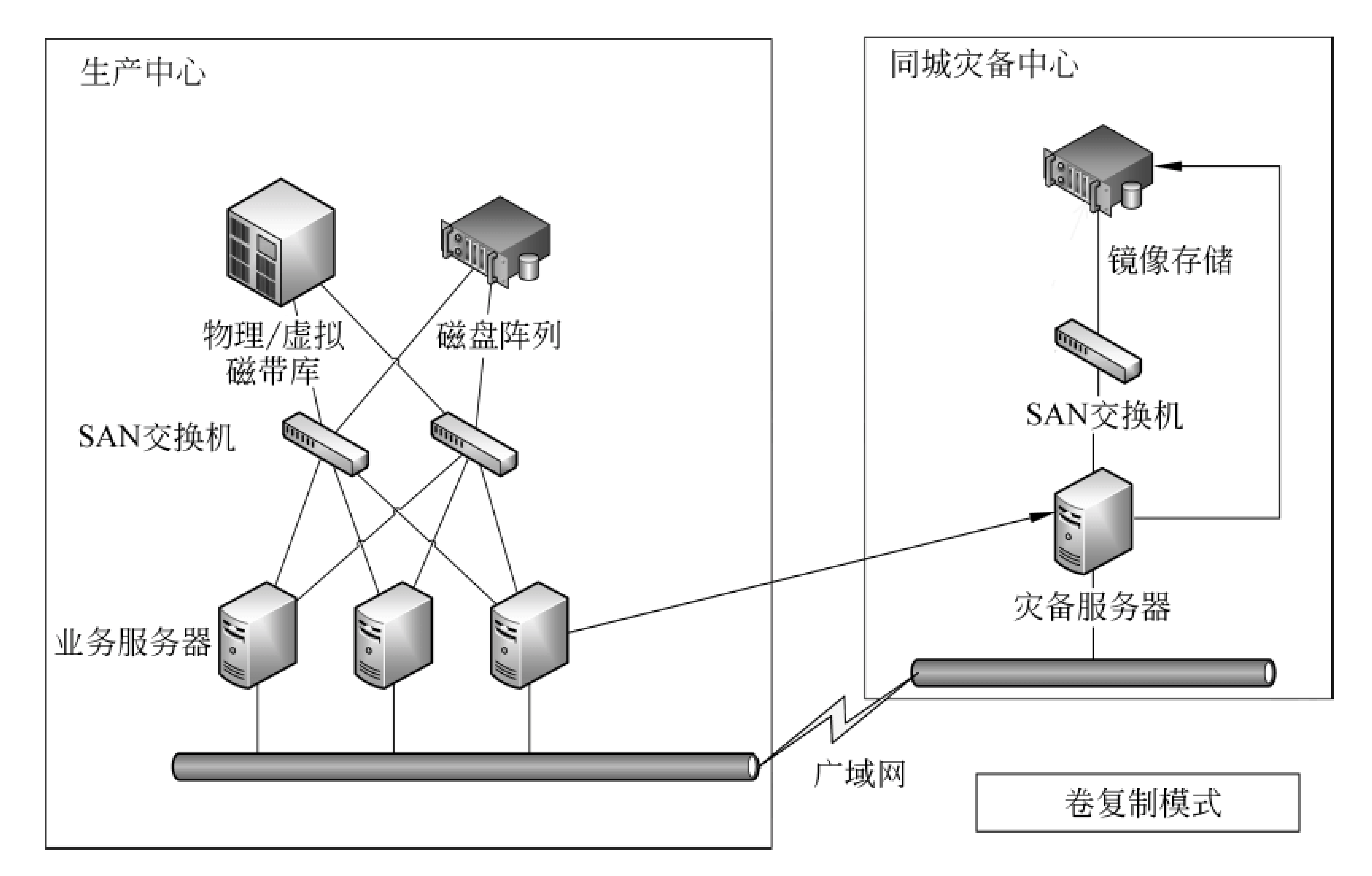


图 6-14　基于数据卷复制示意图(1)

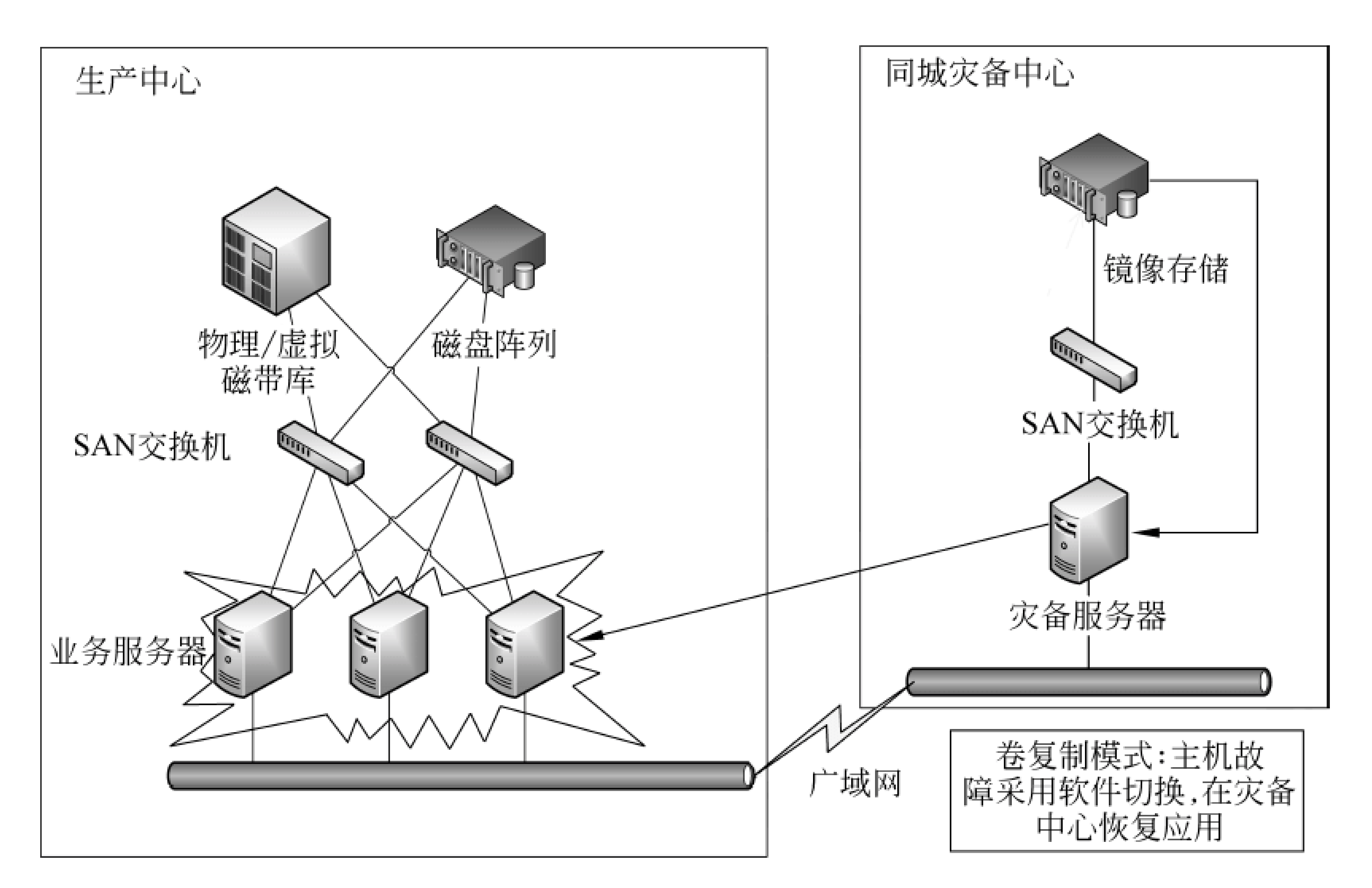


图 6-15　基于数据卷复制示意图(2)

(5) 如果需要灾难自动切换功能，则需要在生产中心和灾备中心的相应服务器上均配置卷管理软件。

(6) 网络带宽需求＝平均数据增量×(峰值持续时间－RPO)÷(峰值数据增量×峰值持续时间－RPO)。

3. 适用范围

数据卷复制灾备解决方案适用于“生产—同城(A—B)，生产—异地(A—B，A—C)”和“生产—同城—异地(A—B—C)及多对一模式”数据卷复制灾备解决方案可以提供“秒级RPO”和“一键式切换”解决方案以保证其业务的连续性。

针对未实现省级集中的社保中心，数据卷复制解决方案可以提供集中式多对一的数据级灾备解决方案，并支持无缝升级至应用级灾备解决方案。

4. 技术特点

(1) 优点

① 秒级RPO：采用数据卷复制灾备解决方案可以实现生产中心数据与灾备中心数据基本一致，在带宽足够的前提下，RPO趋近于0。

② 在配置集群管理软件的前提下，灾难切换时间短，切换操作简易：在主机发生故障或整个生产中心失效时，可以采用专业存储管理软件的"一键式切换"功能，将生产中心的业务迅速自动切换至灾备中心，以保证业务的连续性。

③ 硬件选择余地较大。与基于智能存储系统的灾备方案不同，数据卷复制灾备方案中生产中心和灾备中心存储系统无须同品牌。

④ 工作在数据卷级别上，对任何的数据库、文件系统和第三方应用均可无缝支持。

⑤ 用于灾备目的的数据平时受到保护，保证关键时刻发挥作用。

⑥ 性能好，只占用系统资源的3%～5%。

⑦ 数据复制技术使用IP网络，对底层的网络设备没有任何特殊要求。

(2) 缺点

灾备中心服务器与生产中心服务器的操作系统需要一致，即在生产中心有几类操作系统，则在灾备中心至少每类操作系统要配置一台主机，同一操作系统的主机可以是多对一模式进行从生产中心到灾备中心的复制。

6.6.3 主要实施步骤

数据卷镜像灾备解决方案的实施相对比较简单，一般无须更改任何硬件环境，而只需要在生产中心和灾备中心安装相应的专业存储管理软件，并配置两中心的存储镜像系统及"一键式切换"系统即可。项目实施的大致步骤如下。

(1) 根据生产系统对数据的时间点要求，进行前期论证。

(2) 对生产系统进行全备份(建议)。

(3) 在生产中心和灾备中心的服务器上安装专业存储管理软件。

(4) 停止业务访问(有停机时间)。

(5) 自动转换数据卷格式。

(6) 建立从生产中心数据卷到灾备中心相应数据卷的复制关系，自动进行初始化数据同步。

(7) 启动业务系统(恢复业务运行)。

(8) 在配置群集软件的情况下，可在线配置"一键式切换"系统。

(9) 利用专业存储管理软件的"仿真模拟演练"功能进行灾备演练(可在线做，无须中断业务、也无须中断灾备系统)。

(10) 通过测试后，投入试运行。

(11) 试运行正常后转入正式运行。

主流技术主要以Symantec Storage Foundation数据卷镜像和卷复制技术为代表。

6.7 基于存储虚拟化的数据复制技术介绍

虚拟存储(Storage Virtualization)就是把多个存储介质模块通过一定的手段集中管理起来,所有的存储模块在一个存储池中得到统一管理,并提供大容量、高速传输功能。

存储虚拟化是将实际的物理存储实体与存储的逻辑表示进行分离,并通过 ZONE 的方式将逻辑卷(或称虚卷)分配给应用服务器,而不用关心其数据是在哪个物理存储实体上。逻辑卷与物理实体之间的映射关系,是可由安装在应用服务器上的卷管理软件(称为主机级虚拟化)、存储子系统的控制器(称为存储级虚拟化)、加入 SAN 的专用装置(称为网络级虚拟化)来控制和管理的。

存储虚拟化复制技术的主要功能是实现生产中心与灾备中心之间的逻辑卷复制,从而屏蔽了两中心物理存储设备的差异。为实现基于逻辑卷的复制,需在生产中心和灾备中心配置虚拟化存储管理装置,该装置可将虚拟化存储网络划分成一个个的虚拟卷,既保证本地应用服务器可以访问,同时装置还可通过生产中心与灾备中心之间的 IP 网络实现虚拟逻辑卷的复制。

按照虚拟化存储管理装置的部署和管理方式可分为三种模式:一是带外数据、带外管理模式;二是带内数据、带外管理模式;三是带内数据、带内管理模式。

6.7.1 模式一:带外数据、带外管理模式

1. 技术描述

虚拟存储管理装置连接在 SAN 网络的旁路,生产中心通过应用服务器或 SAN 交换机上的虚拟化端口对写入的生产数据进行拆分,在应用数据进行写操作时,应用服务器或 SAN 交换机上的虚拟化端口截取这些写入操作,并把该写入操作在继续其正常写入物理存储系统的同时并行地复制到本地的虚拟化存储管理装置上,虚拟化存储管理装置在接收到上述数据之后进行压缩打包,并通过 SAN 路由器的 IP 端口复制到灾备中心的虚拟化存储装置中。

对虚拟存储管理装置的管理可通过 IP 方式进行,虚拟存储管理装置发生故障时,不影响应用服务器对物理存储设备的访问,管理平台对该装置的管理不影响生产业务系统的性能,如图 6-16 所示。

2. 资源配置要求

(1) 生产中心与灾备中心通过专线进行连接。

(2) 生产中心与灾备中心可采用异构存储设备。

(3) 生产中心和灾备中心配置虚拟化存储管理装置,连接在各中心 SAN 交换机的旁路上;并在虚拟化存储管理装置上配置虚拟卷复制软件。

(4) 虚拟化存储管理装置的管理平台连接在本地 IP 网络交换机上,通过 IP 网实现对该装置的管理。

3. 技术特点

(1) 虚拟化存储管理装置通过旁路接入生产中心和灾备中心的 SAN 中,生产数据

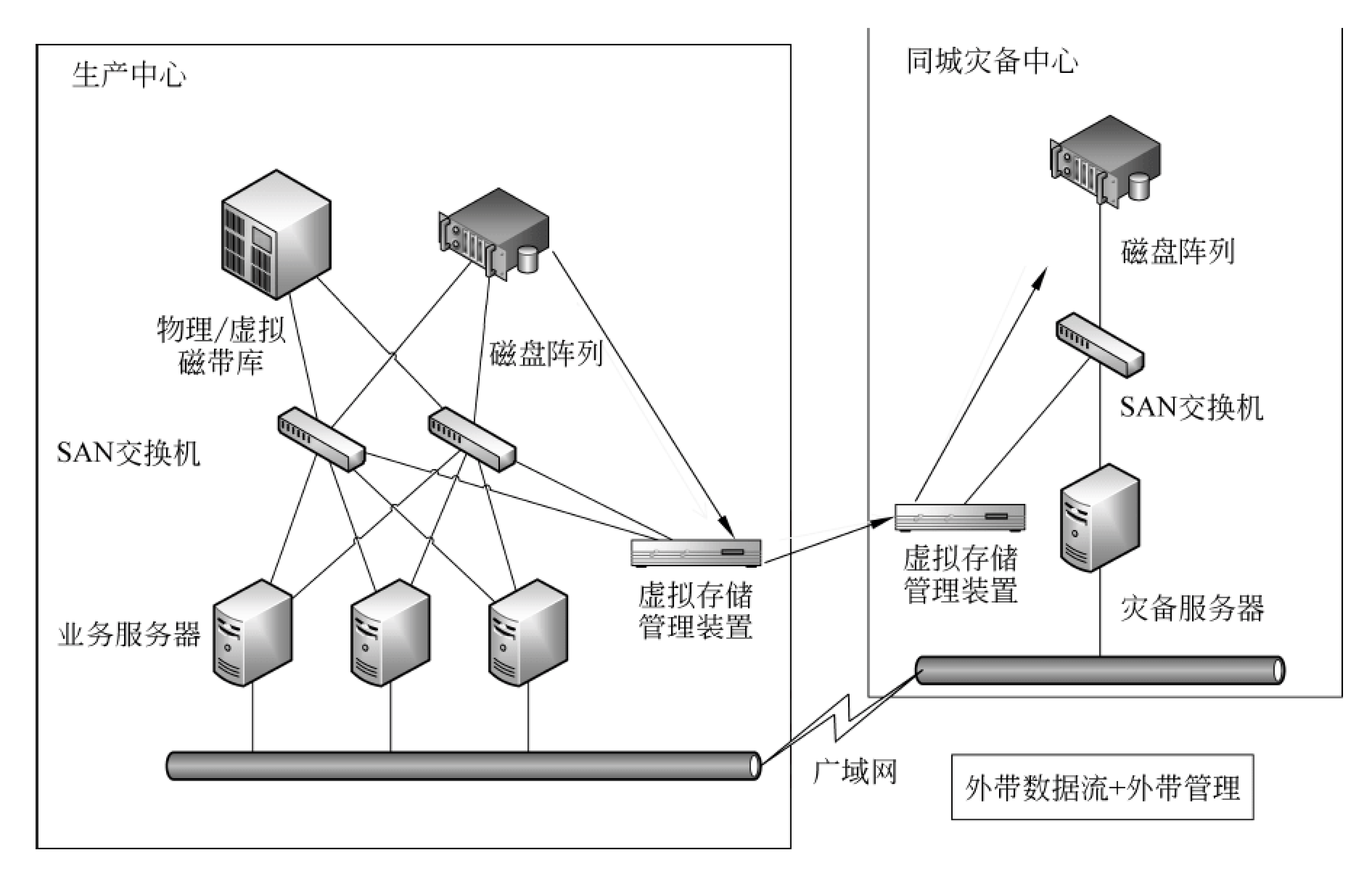


图 6-16　基于存储虚拟化的数据复制示意图(1)

可通过应用服务器或 SAN 交换机的端口进行拆分，该装置的故障不会影响生产业务系统。

(2) 对虚拟化存储管理装置的管理通过 IP 方式实现，不占用 SAN 网络的资源，不影响生产数据复制的性能。

(3) 复制到灾备中心的逻辑卷可直接被灾备中心备份应用服务器使用。

(4) 支持多点虚拟化复制，即多个生产中心的异构存储向同一个灾备中心集中存储进行复制。

4. 主流技术

主流技术为 EMC Recover Point、Falcon NSS/CDP。

6.7.2　模式二：带内数据、带外管理模式

1. 技术描述

虚拟存储管理装置连接在 SAN 网络中，应用服务器在 SAN 网络只能通过虚拟存储管理装置访问虚拟卷。应用服务器通过虚拟存储管理装置对物理存储进行访问，同时虚拟存储管理装置可将生产中心的逻辑卷通过网络镜像到灾备中心的虚拟存储管理装置中。

对虚拟存储管理装置的管理可通过 IP 方式进行，虚拟存储管理装置发生故障时，不影响应用服务器对物理存储设备的访问，管理平台对该装置的管理不影响生产业务系统的性能，如图 6-17 所示。

2. 资源配置要求

(1) 生产中心与灾备中心通过专线进行连接。

(2) 生产中心与灾备中心可采用异构存储设备。

(3) 生产中心和灾备中心配置虚拟化存储管理装置，连接在各中心的 SAN 交换机上，

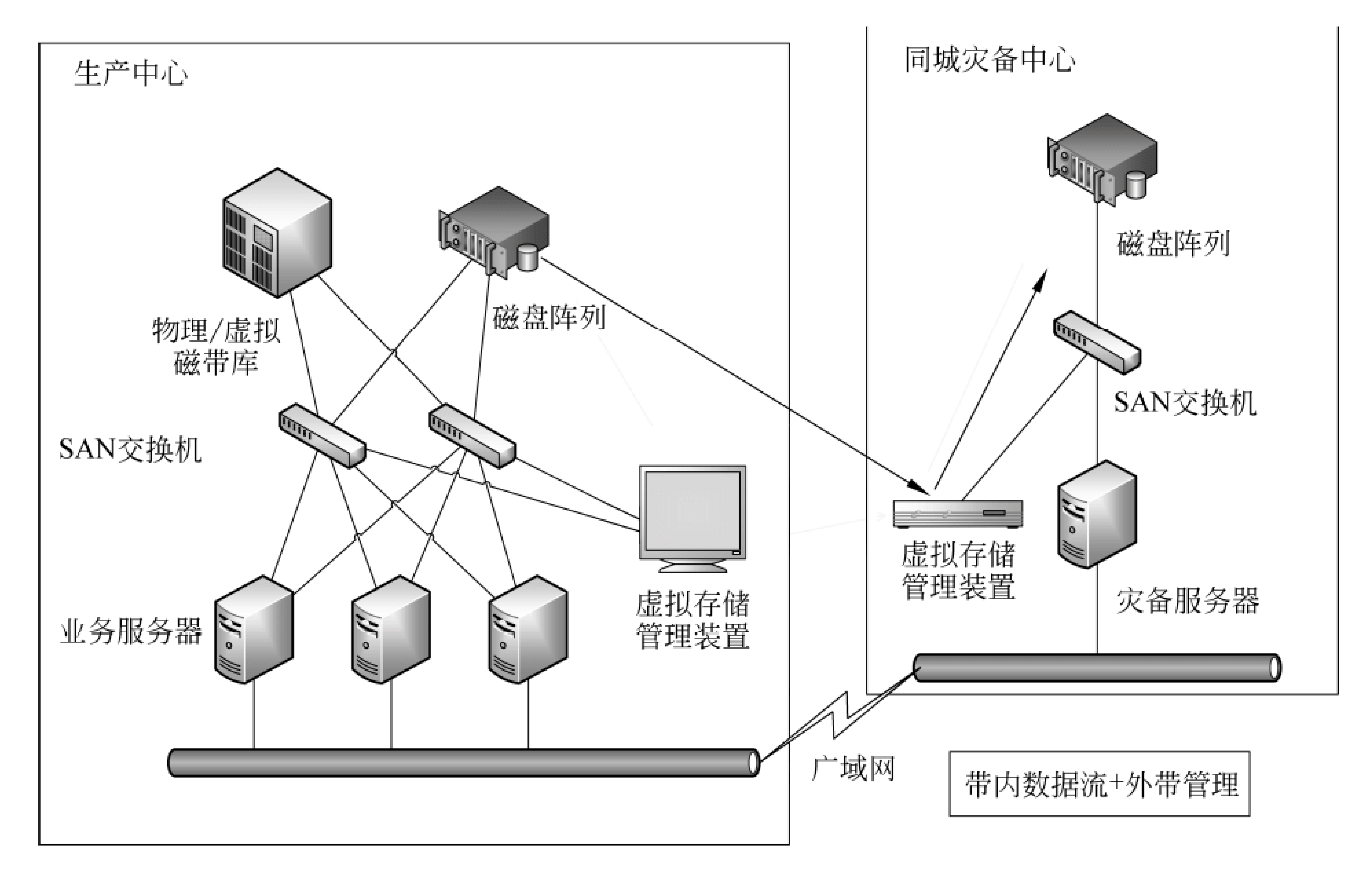


图 6-17　基于存储虚拟化的数据复制示意图(2)

并在虚拟化存储管理装置上配置虚拟卷管理及卷镜像软件。

(4) 虚拟化存储管理装置的管理平台连接在本地 IP 网络交换机上,通过 IP 网实现对该装置的管理。

3. 技术特点

(1) 虚拟化存储管理装置接入生产中心和灾备中心的 SAN 中,由于应用服务器通过该装置进行数据的读写和逻辑卷复制,因此该装置的故障不会影响生产业务系统。

(2) 对虚拟化存储管理装置的管理通过 IP 方式实现,不占用 SAN 网络的资源,不影响生产数据复制的性能。

(3) 复制到灾备中心的逻辑卷可直接被灾备中心备份到应用服务器中使用。

(4) 支持多点虚拟化复制,即多个生产中心的异构存储向同一个灾备中心集中存储进行复制。

4. 主流技术

主流技术为 IBM San Volume Controller、LSI SVM。

6.7.3　模式三:带内数据、带内管理模式

1. 技术描述

虚拟存储管理装置连接在 SAN 网络中,应用服务器通过 SAN 网络只能通过虚拟存储管理装置访问虚拟卷。虚拟存储管理装置通过 SAN 网络进行管理,虚拟存储管理装置发生故障时,会影响正常的生产业务。在这种方式下,虚拟存储管理装置既用于应用服务器对虚拟卷的访问,也可实现生产中心与灾备中心之间的复制。同时对虚拟存储管理装置的管理将影响生产业务和数据复制的性能。

对虚拟存储管理装置的管理可通过 IP 方式进行，虚拟存储管理装置发生故障时，不影响应用服务器对物理存储设备的访问，管理平台对该装置的管理不影响生产业务系统的性能如图 6-18 所示。

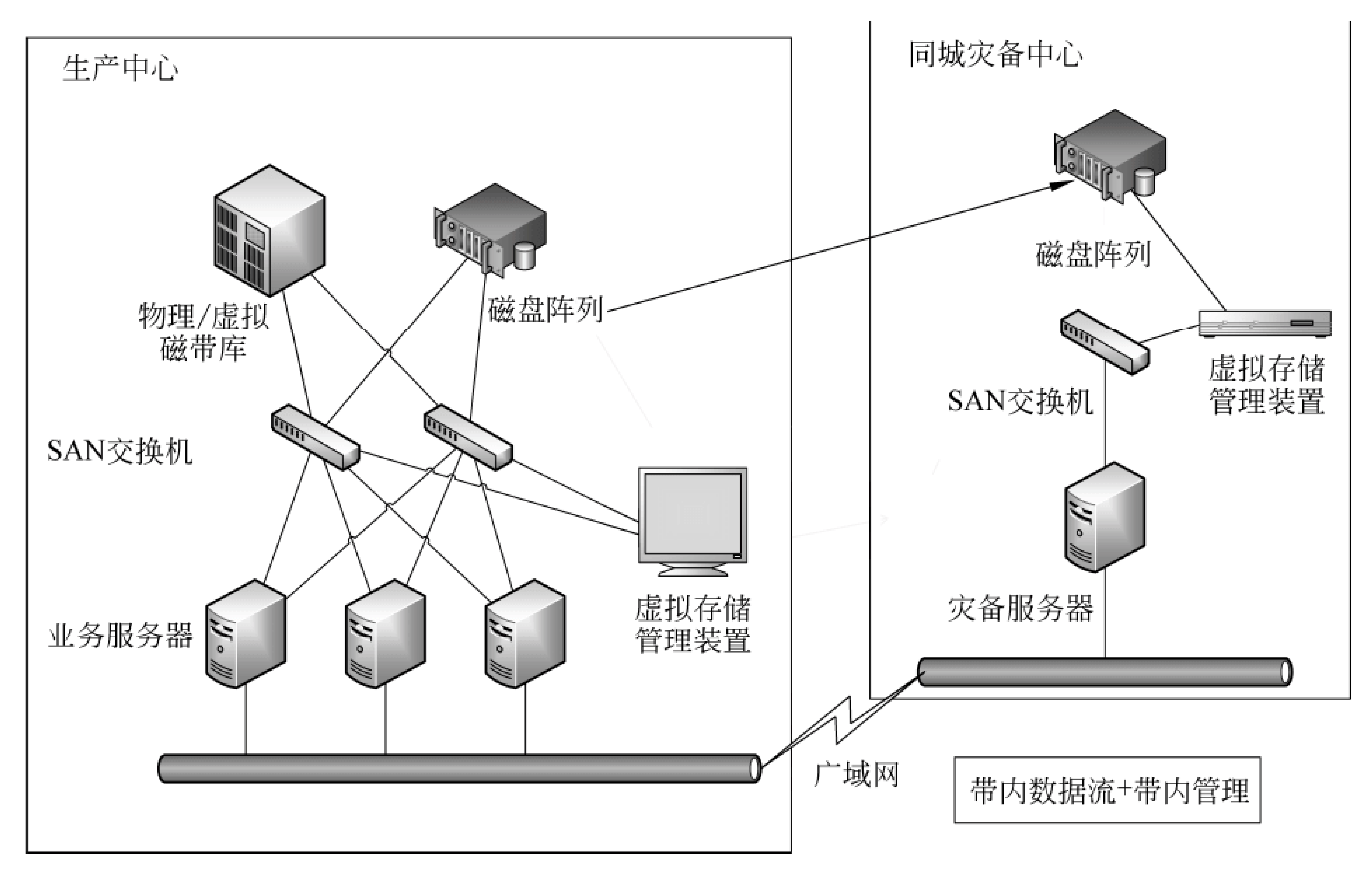


图 6-18 基于存储虚拟化的数据复制示意图(3)

2. 资源配置要求

(1) 生产中心与灾备中心通过专线进行连接。

(2) 生产中心与灾备中心可采用异构存储设备。

(3) 生产中心和灾备中心配置虚拟化存储管理装置，连接在各中心的 SAN 交换机上，并在虚拟化存储管理装置上配置虚拟卷管理及卷镜像软件。

(4) 虚拟化存储管理装置的管理平台连接在 SAN 网络交换机上，通过 SAN 网实现对该装置的管理。

3. 技术特点

(1) 虚拟化存储管理装置接入生产中心和灾备中心的 SAN 中，由于应用服务器通过该装置进行数据的读写和逻辑卷复制，因此该装置的故障不会影响生产业务系统。

(2) 对虚拟化存储管理装置的管理通过 SAN 方式实现，占用 SAN 网络的资源，对生产数据复制的性能有影响。

(3) 复制到灾备中心的逻辑卷可直接被灾备中心备份应用服务器使用。

(4) 支持多点虚拟化复制，即多个生产中心的异构存储向同一个灾备中心集中存储进行复制。

4. 主流技术

主流技术为 DataCore。

6.8 灾备技术对比表

灾备技术对照表见表 6-1。

表 6-1 灾备技术对比表

特点	数据复制技术				数据备份技术
	基于存储虚拟化	基于主机	基于智能存储设备	基于数据库	
实现方式	基于虚拟化存储管理装置的实时数据复制技术	基于逻辑卷的数据实时复制技术	基于磁盘系统的实时复制技术	基于数据库LOG的复制技术	基于备份软件的定时备份
系统架构	在生产中心和灾备中心的SAN网络中部署虚拟化存储管理装置	在生产中心和灾备中心的应用服务器上安装复制软件	在生产中心和灾备中心的磁盘阵列上部署数据复制软件	在生产中心和灾备中心部署数据库复制软件	在生产中心或灾备中心部署备份服务器及备份软件
主机资源	数据复制不占用主机资源，对主机访问存储性能有一定的影响	数据复制占用主机资源	数据复制不占用主机资源	数据复制占用一定的主机资源	数据备份对主机资源占用较少
支持模式	异步、定点、多点	同步、异步、定点、多点	同步、异步、定点	异步、定点、多点	定点
RPO	0 至数分钟	0 至数分钟	0 至数分钟	数分钟至 1 小时	1 天至数天
RTO	低于 1 小时	低于 30 分钟	低于 1 小时	低于 30 分钟	数小时至 1 天
复制数据范围	数据库、文件	数据库、文件	数据库、文件	数据库	文件
网络带宽	对带宽要求较低	卷镜像对带宽要求较高，卷复制对带宽要求低	对带宽要求高	对带宽要求较低	对带宽要求较低（根据复制频率而定）
对原系统的影响	需要在应用服务器上安装虚拟卷管理软件	需要在应用服务器上安装卷管理软件	需要磁盘数据复制软件及协议转换设备	复制软件部署在数据库服务器上	在应用服务器上安装代理服务器
系统切换	系统切换相对简单	结合相关的广域集群软件，可实现系统的自动切换和接管	存储系统切换灵活	切换复杂，需要熟悉数据库工程师	需要进行备份数据的恢复，切换时间很长
实施难度	实施工作有一定难度，对生产系统有一定影响	实施工作有一定难度，对生产系统有一定影响	实施过程复杂	实施工作有一定难度，对生产系统有一定影响	实施工作对生产系统影响最小

任务拓展

1. 什么是结构化和非结构化数据？
2. 简述以信息为中心的存储架构相对于以服务器为中心的架构的优势。
3. 虚拟化的数据中心与传统数据中心相比有哪些优势？
4. 调查存储和管理非结构化数据面临的问题。

第 7 章　虚拟化技术



任务目标

- 了解什么是虚拟化；
- 了解虚拟化技术的架构分类；
- 了解主流虚拟化技术；
- 了解虚拟化的问题和发展趋势。



项目背景

随着计算机硬件和应用的高速发展，硬件和软件的性能不匹配度越来越高，通常普通应用无法消耗服务器资源，造成硬件资源大量浪费。而部分大型应用对资源的需求又无法由单台服务器满足。随着这种矛盾的不断加剧，虚拟化技术的更新和发展已变得刻不容缓。



项目描述

随着聚比特科技有限公司业务的不断发展，需要的应用系统越来越多，而为这些应用系统搭建的服务器也快速增长。导致机房逐渐臃肿，不仅导致机房维护难度越来越大，而且公司为服务器等硬件设备的采购、耗材、能源消耗等投入的资金也越来越多。



项目分析

聚比特科技有限公司技术部面对公司的困难，开始考虑如何更有效地利用自身的硬件资源，减少资源浪费，降低公司运营成本。经过了解，他们发现虚拟化系统非常符合公司现阶段面临的情况。



项目实现

聚比特科技有限公司通过虚拟化建设，大大地提高了硬件的利用率，降低了管理难度，并且可以淘汰部分老旧设备，减少机房的安全隐患。

7.1　虚拟化技术概述

7.1.1　虚拟化的定义

虚拟化（Virtualization）是一个广义的术语，简单来说，是指计算机相关模块在虚拟的基

础上而不是真实、独立的物理硬件基础上运行，这种把有限的固定资源根据不同需求进行重新规划以达到最大利用率，从而实现简化管理、优化资源等的解决方案，叫作虚拟化技术。虚拟化使用软件的方法重新定义并划分 IT 资源，可以实现 IT 资源的动态分配、灵活调度、跨域共享，提高 IT 资源利用率，使 IT 资源能够真正成为社会基础设施，如图 7-1 所示，满足各行各业中灵活多变的应用需求。

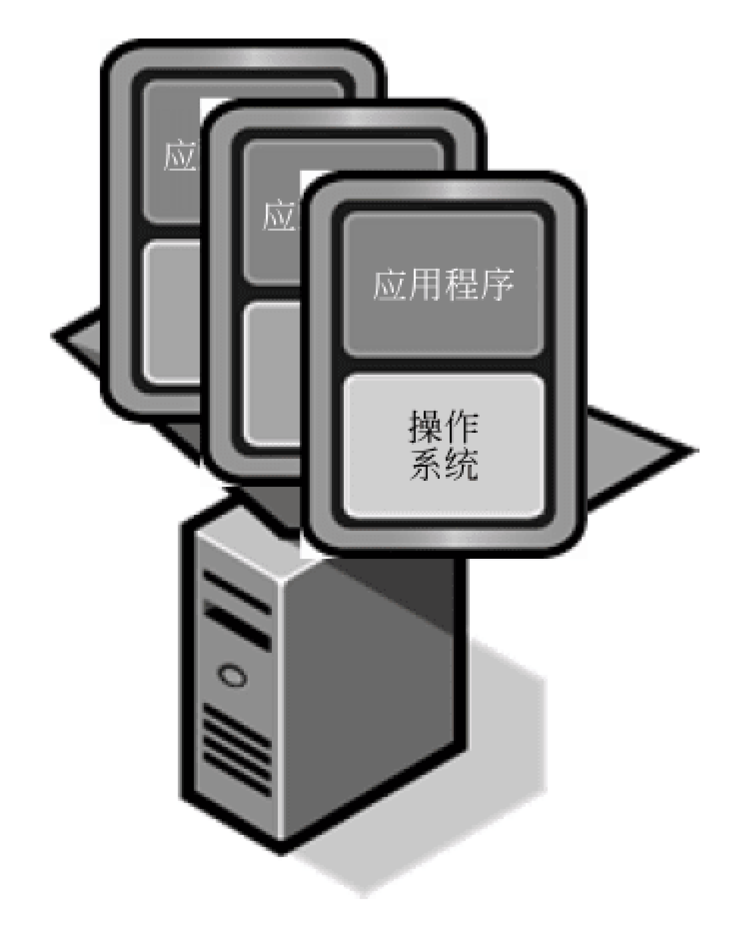


图 7-1　虚拟技术示意图

以下是一些业界标准组织对虚拟化的定义。

“虚拟化是以某种用户和应用程序都可以很容易从中获益的方式来表示计算机资源的过程，而不是根据这些资源的实现、地理位置或物理包装的专有方式来表示它们。换句话说，它为数据、计算能力、存储资源以及其他资源提供了一个逻辑视图，而不是物理视图。”（Jonathan Eunice，Illuminata Inc）

“虚拟化是表示计算机资源的逻辑组（或子集）的过程，这样就可以用从原始配置中获益的方式访问它们。这种资源的新虚拟视图并不受现实、地理位置或底层资源的物理配置的限制。”（Wikipedia）

“虚拟化：对一组类似资源提供一个通用的抽象接口集，从而隐藏属性和操作之间的差异，并允许通过一种通用的方式来查看并维护资源。”（Open Grid Services Architecture Glossary of Terms）

虚拟化概念并不是新概念。早在 20 世纪 70 年代，大型计算机就一直在同时运行多个操作系统实例，每个实例也彼此独立。直到最近，软硬件方面的进步才使得虚拟化技术逐渐出现在基于行业标准的中低端服务器上。

然而，虚拟化技术的内涵远远不止于虚拟服务器。目前，已经有了网络虚拟化、CPU 虚拟化和存储虚拟化等技术。如果在一个更广泛的环境中或从更高级的抽象角度来思考虚拟化技术，虚拟化技术就变成了一个非常强大的概念，可以为最终用户、应用程序和企业提供很多帮助。

7.1.2　虚拟化的发展历史

1. 虚拟化技术的萌芽

20 世纪 60 年代开始，美国计算机学术界就有了虚拟技术思想的萌芽。1959 年克里斯托弗（Christopher Strachey）发表了一篇学术报告，名为《大型高速计算机中的时间共享》（*Time Sharingin Large Fast Computers*），他在报告中提出了虚拟化的基本概念，这篇文章也被认为是虚拟化技术的最早论述。

L. W. Comeau 和 R. J. Creasy 创造性地设计了一种名为 CP-40 的新型操作系统，该操作系统实现了虚拟内存和虚拟机。

2. 20 世纪六七十年代：虚拟化技术在大型机上的实现

虚拟化技术在 20 世纪 60 年代首次出现，由 IBM 率先实施：是一种对大型计算机进行逻辑分区以形成若干独立虚拟机的方式。这些分区允许大型计算机进行“多任务处理”，即

同时运行多个应用程序和进程。原因是当时大型计算机是十分昂贵的资源，因此设计了虚拟化技术来进行分区，作为一种充分利用投资的方式，解决了大型计算机的僵化和使用率不足的问题。

1965 年 8 月，IBM 推出 System/360 Model 67 和 TSS 分时共享系统（Time Sharing System），通过虚拟机监视器（Virtual Machine Monitor）虚拟所有的硬件接口，允许很多远程用户在同一时间共享同一高性能计算设备。在 IBM 内部，Model 67 与另一个被称为 CP-67 的系统配合使用，以保证使用一台 360/67 模仿多台不同型号的计算机。同年还发布了 M44/44X 计算机项目，定义了虚拟内存管理机制，用户程序可以运行在虚拟的内存中，对于用户来说，这些虚拟内存就好像一个个虚拟机，为多个用户的程序提供了独立的计算环境。

1972 年，IBM 发布用于创建灵活大型主机的虚拟机（VM）技术，该技术可根据动态的需求快速而有效地使用各种资源。一批新的拥有虚拟化功能的产品涌现出来，这些机器在当时都具有虚拟机功能，通过一种叫作 VMM（虚拟机监控器）的技术在物理硬件之上生成了很多可以运行独立操作系统软件的虚拟机实例。

3. 20 世纪八九十年代：虚拟化技术在小型机上的实现

在 20 世纪八九十年代，由于客户端—服务器应用程序以及价格低廉的 x86 服务器和台式机组成了分散的计算机架构，大型计算机上的虚拟化技术处于停滞不前的状态。由于虚拟化技术在商业应用上的优势，RISC 服务器与小型计算机成为虚拟化技术第二代受益者。IBM 公司在 AS/400 上提出上逻辑分区（LPAR，如图 7-2 所示）技术和新的高可行性集群解决方案。在电源管理程序上运行的 AS/400LPAR 令单台服务器工作起来如同 12 个独立的服务器。随后这项技术还应用在 IBM RS/6000 服务器（后来的 pSeries 服务器）上。

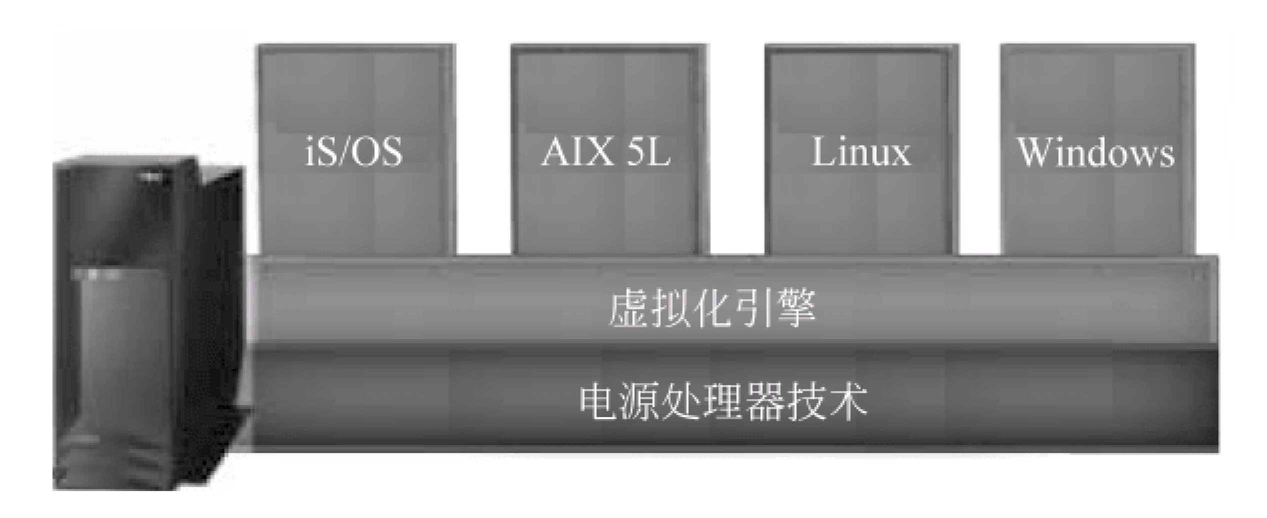


图 7-2　LPAR 技术

4. 21 世纪：虚拟化技术在 x86 服务器上的实现

20 世纪 90 年代 Windows 的广泛使用以及 Linux 作为服务器操作系统的出现奠定了 x86 服务器的行业标准地位。x86 服务器和桌面部署的增长带来了新的 IT 基础架构和运行难题。这些难题包括以下方面。

(1) 基础架构利用率低

根据市场调研公司美国国际数据集团（International Data Corporation，IDC）的报告，典型的 x86 服务器部署平均达到的利用率仅为总容量的 10%～15%。组织通常在每台服务器上运行一个应用程序，以避免出现一个应用程序中的漏洞影响同一服务器上其他应用程序的可用性风险。

(2) 物理基础架构成本日益攀升

为支持不断增长的物理基础架构而需要的运营成本稳步攀升。大多数计算基础架构都必须时刻保持运行,因此耗电量、制冷和设施成本不随利用率水平而变化。

IT 管理成本不断攀升。随着计算环境日益复杂,基础架构管理人员所需的专业教育水平和经验以及使用此类人员的相关成本也随之增加。组织在与服务器维护相关的手动任务方面花费过多的时间和资源,因而也需要更多的人员来完成这些任务。

(3) 故障切换和灾难保护不足

关键服务器应用程序停机和关键最终用户桌面不可访问对组织造成的影响越来越大。安全攻击、自然灾害、流行疾病以及恐怖主义的威胁使得对桌面和服务器进行业务连续性规划显得更为重要。

(4) 最终用户桌面的维护成本高昂

企业桌面的管理和保护带来了许多难题。在不影响用户有效工作的情况下控制分布式桌面环境并强制实施管理、访问和安全策略,实现起来十分复杂且成本高昂。必须不断地对桌面环境应用数目众多的修补程序和升级以消除安全漏洞。

在早期,由于 x86 计算机性能的低下和推广程度有限,以上问题并未突出。随着技术的发展,以上矛盾日益尖锐,基于 x86 体系结构的计算机开始出现了 20 世纪 60 年代大型计算机经历过的同样问题,即缺乏灵活性和未得到充分利用以及上述提到的一系列问题。

针对以上问题,VMware 公司在 1999 年推出了针对 x86 系统的虚拟化技术,旨在解决上述很多难题,并将 x86 系统转变成通用的共享硬件基础架构,以便使应用程序环境在完全隔离、移动性和操作系统方面有选择的空间。

7.1.3　百花齐放的虚拟化技术

在 20 世纪,虚拟化技术基本上都是服务器虚拟化,进入了 21 世纪,随着 IT 的发展,虚拟化的思路被借用到服务器以外的领域(包括存储、网络、桌面应用等),形成了各种各样的虚拟化技术。

1. 存储虚拟化技术

所谓虚拟存储技术,就是把各种不同的存储设备有机地结合起来使用,从而得到一个容量很大的“存储池”,可以提供给各种服务器灵活使用,并且可以在各存储设备间灵活转移数据,称为虚拟存储。

存储虚拟化的基本概念是将实际的物理存储实体与存储的逻辑表示分离开来,应用服务器只与分配给它们的逻辑卷(或称虚卷)打交道,而不用关心其数据是在哪个物理存储实体上。逻辑卷与物理实体之间的映射关系是由安装在应用服务器上的卷管理软件(称为主机级的虚拟化),或存储子系统的控制器(称为存储子系统级的虚拟化),或加入存储网络 SAN 的专用装置(称为网络级的虚拟化)来管理的。

主流的虚拟存储技术厂商和产品有 EMC 的 Invista、IBM 的 SVC、HDS 的 UPS 等。

2. 服务器虚拟化技术的进一步发展:

在历史悠久的服务器硬件虚拟化方面,在 RISC 领域 IBM 更进一步,其 2002 年发布的 AIX5L v5.2 首次包括了 IBM 实现的动态逻辑分区(DLPAR)。DLPAR 允许在无须重启系统的情况下,将包括处理器、内存和其他组件在内的系统资源分配给独立的分区。这种在不

中断运行的情况下进行资源分配的能力不仅令系统管理变得更加轻松，而且因为能够更好地使用资源而帮助降低总成本。

3. 应用虚拟化技术：

在桌面应用来看，开始出现了应用虚拟化(也称为桌面虚拟化)的技术，该技术把应用程序的人机交互逻辑(应用程序界面、键盘及鼠标的操作、音频输入输出、读卡器、打印输出等)与计算逻辑隔离开来，客户端无须安装软件，通过网络连接到应用服务器上，计算逻辑从本地迁移到后台的服务器完成，实现应用的快速交付和统一管理，如图 7-3 所示。

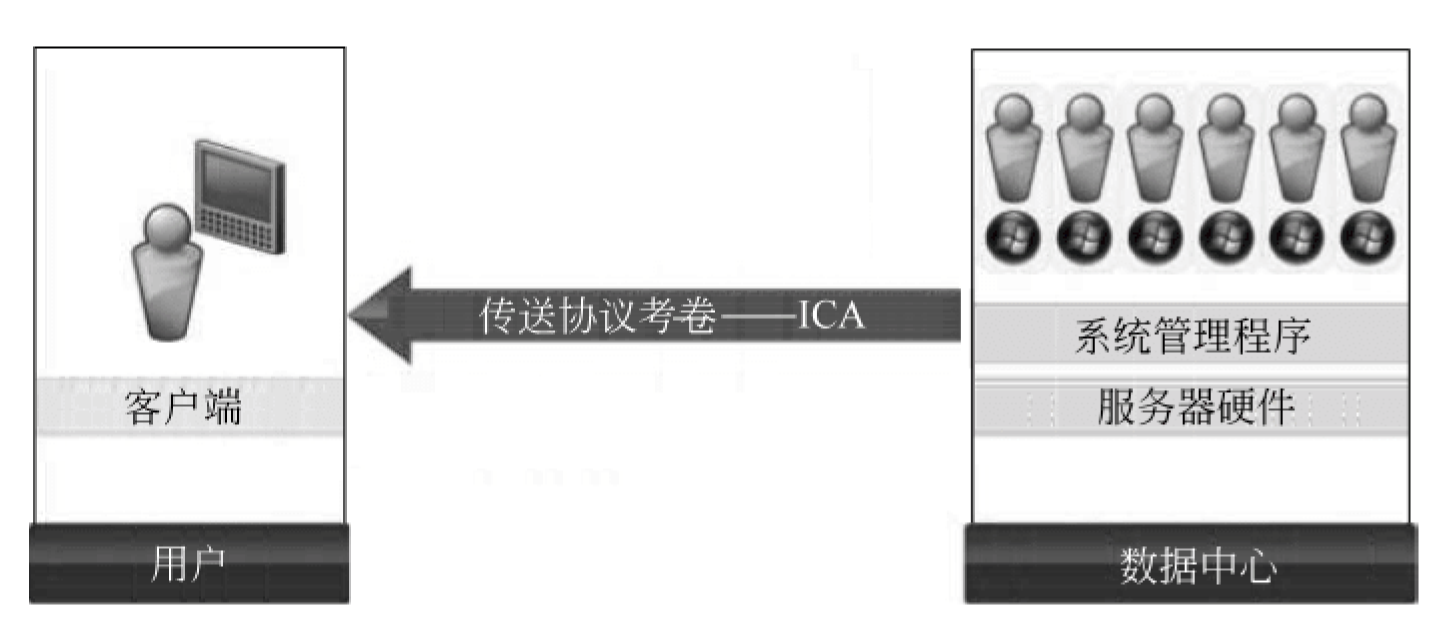


图 7-3　虚拟桌面示意图

4. 网络虚拟化技术

在网络方面，虚拟设备(如虚拟交换)的出现就是网络虚拟化最好的体现。企业网络管理者通过在交换机上开启虚拟交换机，从而实现虚拟化共享 IT 资源并将它们按需求分配给特定的任务，使用户可以用更少的物理资源满足应用需求。同时很多网络技术，诸如帧中继、逻辑分区和 RAID 等都可使用虚拟化。虚拟化正在通过新的方式被用于数据中心，使网络功能得以巩固自动完成。

对以上各种虚拟化技术在后文中将会详细介绍。

5. VCE 联盟

值得一提的是，2009 年 11 月 9 日，VMware、Cisco、EMC 三家共同出资，Intel 公司参股的 Acadia 公司正式成立，三家主要合作公司建立了虚拟计算环境联盟 VCE(Virtual Computing Environment)，当然这个联盟的首字母缩写也是这三个 IT 公司名称的首字母，并推出了类似于虚拟化架构基础软件包的 Vblock 基础架构包，其整合了 Cisco 的统一计算系统(UCS)和交换机，EMC 的存储和安全技术以及 VMware 的虚拟化技术。

Vblock 基础架构包是一个完全集成、经过测试和验证的、整装待发和蓄势成长的基础架构包，它融合了 Cisco、EMC 和 VMware 提供的业内最强的虚拟化、网络、计算、存储、安全和管理技术，并且具备端到端的供应商可靠性。通过 Cisco 的统一计算系统、EMC 的虚拟化数据中心概念机存储系统以及控制着绝大部分 x86 环境下虚拟化应用的 VMware 虚拟化产品，将通过 Vblock 基础架构包(如图 7-4 所示，这是“虚拟计算环境联盟”所推出的产品)，以整合的概念推广到用户中。这三家虚拟化巨头的结盟，将会给虚拟化产业带来深远的影响。

7.1.4　虚拟化的优势

与传统 IT 资源分配的应用方式相比，虚拟化有以下优势。



图 7-4　Vblock 基础架构包

（1）虚拟化技术可以大大提高资源的利用率

为了达到资源的最高利用率，虚拟化把一个硬件虚拟成多个硬件，这里的一个硬件指的不是一个个体，而是由多个个体组成的一组资源，例如可以将多个硬盘组成阵列，将多个硬盘视为计算机的硬盘部分。用户将许多资源组成一个庞大的、计算能力十分巨大的“巨型计算机”，再将这个巨型计算机虚拟成多个独立的系统，这些系统相互独立，但共享资源，这就是虚拟化的精髓。传统的 IT 企业为每一项业务应用部署一台单独的服务器，服务器的规模通常是针对峰值配置，服务器规模（处理能力）远远大于服务器的平均负载，服务器在大部分时间处于空闲状态，资源得不到最大利用。虚拟化技术能够动态调用空闲资源，减小服务器规模，提高资源利用率。

（2）提供相互隔离、安全、高效的应用执行环境

用户可以在一台计算机上模拟多个系统及多个不同的操作系统。虚拟系统下的各个子系统相互独立，即使一个子系统遭受攻击而崩溃，也不会对其他系统造成影响，而且在使用备份机制后，子系统可以被快速地恢复。同时，应用执行环境简单易行，大大提高了工作效率，降低了总体的投资成本。

（3）采用虚拟化技术后，虚拟化系统能够方便地管理和升级资源

传统的 IT 服务器资源是硬件相对独立的个体，对每一个资源都要进行相应的维护和升级，会耗费企业大量的人力和物力。虚拟化系统将资源整合，在管理上十分方便，在升级时只需添加动作，避开传统企业进行容量规划、定制服务器、安装硬件等工作，从而提高了工作效率。

7.1.5　虚拟化的目的

虚拟化的主要目的是对 IT 基础设施进行简化。它可以简化对资源以及对资源管理的访问。

消费者可以是一名最终用户、应用程序、访问资源或与资源进行交互的服务。资源是一个提供一定功能的实现，它可以基于标准的接口接受输入和提供输出。资源可以是硬件，例如服务器、磁盘、网络、仪器；也可以是软件，例如 Web 服务。

虚拟化支持的操作系统有 Windows 和 Linux 各种系统。

消费者通过受虚拟资源支持的标准接口对资源进行访问。使用标准接口，可以在 IT 基础设施发生变化时将对消费者的破坏降到最低。例如，最终用户可以重用这些技巧，因为

他们与虚拟资源进行交互的方式并没有发生变化，即使底层物理资源或实现已经发生了变化，他们也不会受到影响。另外，应用程序也不需要进行升级或应用补丁，因为标准接口并没有发生变化。

IT 基础设施的总体管理也可以得到简化。因为虚拟化降低了消费者与资源之间的耦合程度，因此，消费者并不依赖于资源的特定实现。利用这种耦合关系，管理员可以在保证管理工作对消费者产生最小影响的基础上实现对 IT 基础设施的管理。管理操作可以手工完成，也可以半自动化完成，或者通过服务级协定(SLA)驱动来自动完成。

在这个基础上，网络计算可以广泛地利用虚拟化技术。网络计算可以对 IT 基础设施进行虚拟化。它处理 IT 基础设施的共享和管理，动态提供符合用户和应用程序需求的资源，同时还将提供对基础设施的简化访问。

7.2 现有虚拟化技术的分析与对比

现有的较成熟的虚拟化技术主要分为服务器虚拟化(也称为操作系统虚拟化)、网络虚拟化、存储虚拟化、应用虚拟化(也称为桌面虚拟化)四种。

7.2.1 服务器虚拟化

服务器虚拟化是虚拟化技术最早细分出来，最成熟的领域。在 2006 年 2 月 Forrester Research 的调查中，全球范围的企业对服务器虚拟化的认知率达到了 75%。1/3 的企业已经在使用或者准备部署服务器虚拟化。这个产生于 20 世纪 60 年代的技术日益显示出其重要价值。由于服务器虚拟化的发展时间长，应用广泛，所以很多时候人们几乎把服务器虚拟化等同于虚拟化。

关于服务器虚拟化的概念，各个厂商有自己不同的定义，然而其核心思想是一致的，即它是一种方法，能够通过区分资源的优先次序并随时随地能将服务器资源分配给最需要它们的工作负载来简化管理和提高效率，从而减少为单个工作负载峰值而储备的资源。

有了虚拟化技术，用户可以动态启用虚拟服务器(又称为虚拟机)，每个服务器实际上可以让操作系统(以及在上面运行的任何应用程序)误以为虚拟机就是实际硬件。运行多个虚拟机还可以充分发挥物理服务器的计算潜能，迅速应对数据中心不断变化的需求。

服务器虚拟化按实现原理来分，主要分为基于 CPU 的虚拟化、基于硬件的虚拟化，以及基于操作系统的虚拟化。

1. 基于 CPU 的虚拟化

在 Intel、AMD 等厂商的设计蓝图中，CPU 虚拟化技术的最终目标是可以单 CPU 模拟多 CPU 并行，允许一个平台同时运行多个操作系统，并且应用程序都可以在相互独立的空间内运行而互不影响，从而显著提高计算机的工作效率。从处理器硬件角度实现了服务器(操作系统)级别虚拟化，从而提高了效率。

在设计蓝图中，支持虚拟技术的 CPU 带有特别优化过的指令集来控制虚拟过程，通过这些指令集，VMM(Virtual Machine Monitor，也就是虚拟机软件)会很容易提高性能，相比软件的虚拟实现方式会在很大程度上提高其性能。虚拟化技术可提供基于芯片的功能，借

助兼容 VMM 软件能够改进纯软件解决方案。由于虚拟化硬件可提供全新的架构，支持操作系统直接在其上运行，从而无须进行二进制转换，减少了相关的性能开销，极大简化了 VMM 设计，进而使 VMM 能够按通用标准进行编写，其性能更加强大。另外，在纯软件 VMM 中，缺少对 64 位客户操作系统的支持，而随着 64 位处理器的不断普及，这一严重缺点也日益突出。而 CPU 的虚拟化技术除支持传统操作系统之外，还支持 64 位客户操作系统。

Intel 自 2005 年末开始便在其处理器产品线中推广应用 Intel Virtualization Technology(Intel VT)虚拟化技术。Intel 已经发布了具有 Intel VT 虚拟化技术的一系列处理器产品，包括桌面平台的 Pentium 4 6x2 系列、Pentium D 9x0 系列和 Pentium EE 9xx 系列，还有 Core Duo 系列和 Core Solo 系列中的部分产品，以及服务器/工作站平台上的 Xeon LV 系列、Xeon 5000 系列、Xeon 5100 系列、Xeon MP 7000 系列以及 Itanium 2 9000 系列；同时绝大多数的 Intel 下一代主流处理器，包括 Merom 核心移动处理器，Conroe 核心桌面处理器，Woodcrest 核心服务器处理器，以及基于 Montecito 核心的 Itanium 2 高端服务器处理器都将支持 Intel VT 虚拟化技术。

而 AMD 方面也已经发布了支持 AMD 虚拟化技术(AMD VT)的一系列处理器产品，包括 Socket S1 接口的 Turion 64 x2 系列以及 Socket AM2 接口的 Athlon 64 x2 系列和 Athlon 64 Fx 系列等，并且绝大多数的 AMD 下一代主流处理器，包括即将发布的 Socket F 接口的 Opteron，都将支持 AMD VT 虚拟化技术。

但虚拟化技术是一套解决方案。完整的情况需要 CPU、主板芯片组、BIOS 和软件的支持，例如虚拟化软件或者某些操作系统本身。但截至目前，这套解决方案还没完全落地和实现。目前 CPU 虚拟化技术暂时还停留在 CPU 对服务器(操作系统)级别虚拟化优化这个层面。

2. 基于硬件的虚拟化

基于硬件的服务器虚拟化产品走得比较靠前的应该是 IBM 和 HP 等服务器硬件的厂商。这两家公司在最新的 RISC 架构服务器及最新版的操作系统中都嵌入了虚拟化技术。

(1) IBM 的硬件虚拟化

IBM 早在 p690 服务器和 AIX 5L 操作系统首次公布的时候，就宣布在其动态逻辑分区(LPAR)技术的支持下，一个系统内可独立地运行多个分区，每个分区运行独立地操作系统，这时候的分区是以 CPU 为“颗粒”的。

到发布 p5 服务器时，IBM 最新的微分区技术打破了分区上以 CPU 为“颗粒”的限制，可以将单个 CPU 划分为 10 个微分区，从而创建和运行比物理处理器数量更多的分区。IBM 同时宣布，新版操作系统 AIX 5.3 可以支持 1/10 CPU 颗粒的微分区。

微分区源自大型主机技术，是基于服务器虚拟化领域的两项主要突破：①物理处理器虚拟化；②I/O 设备虚拟化。两项虚拟化都可实现分区对这类计算资源的共享。可实现以下 3 大功能：①可精细调整的资源分配；②更大分区数量的承载能力；③更高资源利用效率。

从成本角度看，用户现有 UNIX 系统的利用率常常只有 10%～25%，一味地通过增加服务器和处理器数量来满足应用，势必使成本上升而收效甚微。常见的分区技术往往是以大于或者等于一颗物理处理器为单位实现的，在实际应用中这一界限有时并不够精细，尤其

是随着单位 CPU 处理能力的不断提高就更是如此。

如果采用微分区技术，效果十分显著。因为这意味着用户在服务器整合中，不再需要买太多的处理器，资源的划分和共享也将更为精细，如图 7-5 所示。

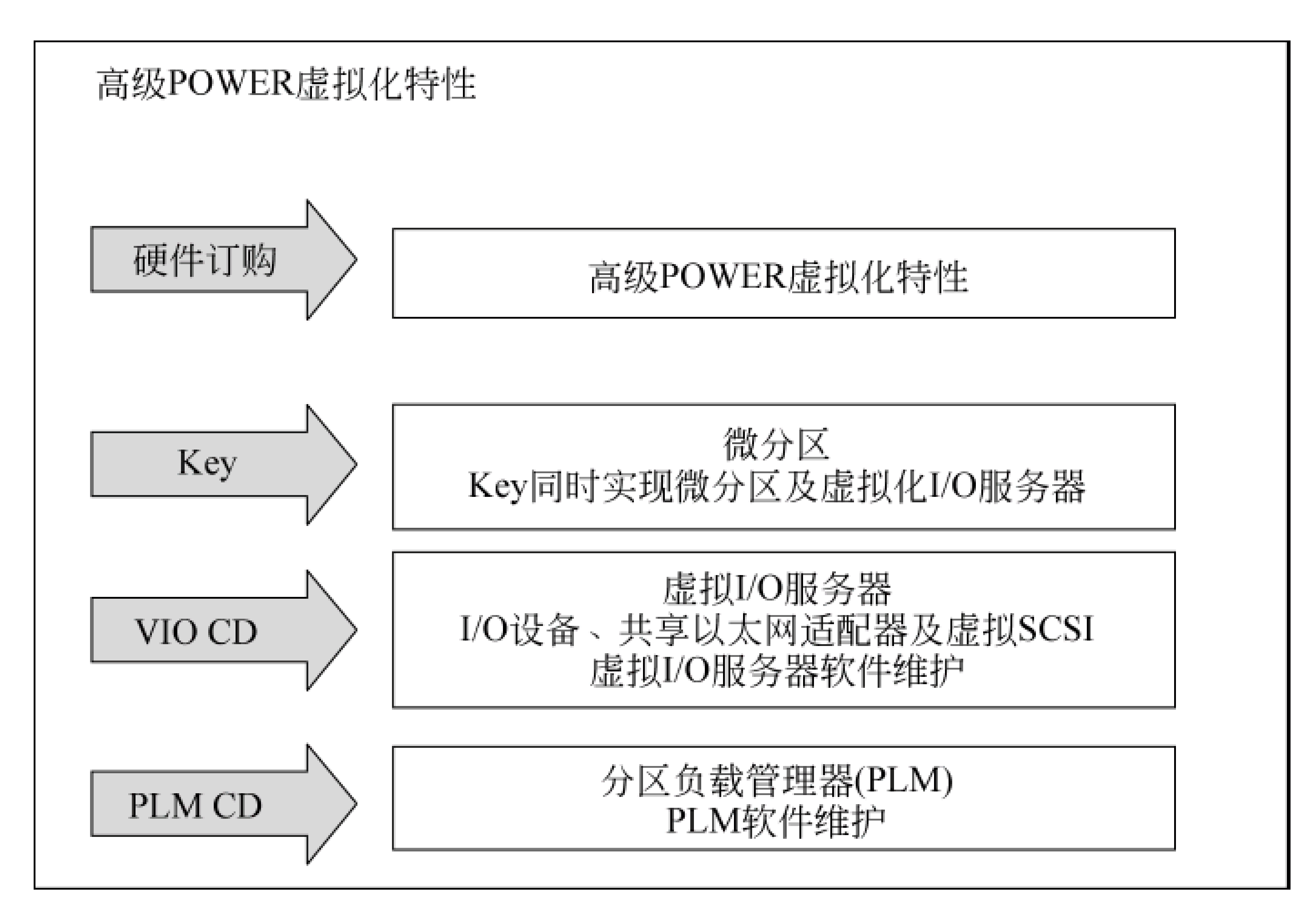


图 7-5　虚拟化微分区技术示意图

后来，IBM 进一步拓展了其服务器虚拟技术的范畴，推出了由操作系统、系统技术和系统服务三部分组成的服务器虚拟引擎。其中，操作系统涉及 AIX、i5/OS、z/OS 和 Linux，其技术宗旨是单台服务器内运行多种操作系统，在异构 IT 基础架构中以统一的方式实现资源的共享和管理以及管理非 IBM 操作系统平台；系统技术包括微分区、VLAN、虚拟 I/O、Hypervisor 等；而系统服务则包括一个服务器系统服务套件和一个存储系统服务套件。在服务器系统服务套件中，包括硬件监督模块 VE Console（虚拟引擎控制台），可以利用两个主要的功能模块 Launchpad 和 Health Center 监控资源的运行状态，进行问题诊断和系统管理；另外还包括硬件管理模块 IBM Direction Multiplatform 整合系统管理。

同时，IBM 还推出了应用虚拟工具套件，包括应用监督模块 Enterprise Workload Management（企业负载管理器，EWLM），可以在异构环境下自动管理分布式企业级系统，根据业务优先级将 IT 服务分类，设立相应的性能目标，并根据这些性能目标提供端到端的性能分析和评估，通过分析，EWLM 自动按照应用拓扑调整网络路由。与 EWLM 相配合的是一个应用管理模块 Tivoli Provisioning Manger（TPM），该模块与 EWLM 配合，可以实现系统部署和配置步骤的自动化，为 IT 系统的自动部署、资源分配和启用提供解决方案。

（2）HP 的硬件虚拟化

除了 IBM 以外，HP 的分区连续技术能够把服务器划分成物理或逻辑独立的分区，为优化资源利用、提高服务器的可用性提供坚实的基础。

① 硬件分区（nPars）：HP nPARS 是单个服务器中的硬件分区。nPARS 根据服务器类型最多提供 16 个硬件分区、完全的硬件和软件隔离能力以及在一个服务器上运行多个 OS 实例。

② 虚拟分区（vPars）：HP vPARS 具有在一个系统或硬件分区内实现执行多个 OS 实例的独特特性。每个 vPARS 能够拥有规定的内存量、一个或多个物理内存区域、一个规定

的 CPU 池、服务器内一个或多个 I/O 卡。能够使用软件命令动态地创立和修改 vPARS。因此，每个应用能够在性能最大、OS 配置要求得到满足的环境中运行。

③ 资源分区：进程资源管理软件(PRM)能够动态地以多种方式把系统资源(CPU、内存和磁盘 I/O)分配给客户的应用，分配的方式可以是根据份额、百分比和处理器组(pSets)。pSets 允许在服务器上创立处理器组，而应用或用户可以被分配到规定的 pSet 上运行。

3. 基于操作系统的虚拟化

基于操作系统的虚拟化允许多个虚拟机通过分配时间和资源带宽的方式，共享同一个底层物理服务器及其资源。由于目前硬件的处理能力一直高于其上运行的应用程序，因此虚拟化允许用户在共享硬件上运行多个应用程序工作负载，借以提高硬件的整体利用率。

采用虚拟化实现服务器整合，除了可以节省总体成本外，还可以提高服务水平和可用性。例如，由于虚拟机是经过封装的，因此可轻易获取和迁移其配置信息和状态信息。虚拟机的虚拟磁盘其实是一些文件，可备份和快速恢复这些文件以备灾难/恢复时使用。

很多情况下“客户”操作系统是通过虚拟机监视器(Virtual Machine Monitor，VMM)来与硬件进行通信，由 VMM 来决定其对系统上所有虚拟机的访问(注意：大多数处理器和内存访问独立于 VMM，只在发生特定事件时才会涉及 VMM，如页面错误等)。在纯软件虚拟化解决方案中，VMM 在软件套件中的位置是传统意义上操作系统所处的位置，而操作系统的位置是传统意义上应用程序所处的位置。这一额外的通信层需要进行二进制转换，以通过提供到物理资源(如处理器、内存、存储、显卡和网卡等)的接口模拟硬件环境。这种转换必然会增加系统的复杂性。此外，客户操作系统的支持受到虚拟机环境的限制，这会阻碍特定技术的部署，如 64 位客户操作系统。在纯软件解决方案中，软件堆栈增加的复杂性意味着这些环境难于管理，因而会加大确保系统可靠性和安全性的困难。

VMware 是服务器虚拟化领域的市场领先产品和创新品牌，提供一套虚拟机解决方案的软件。除了 VMware 之外，业界还有微软 Hyper-V 和 Xen 等服务器虚拟化软件。

Xen 是一种著名的开放源代码的虚拟化技术，它基于 Linux 平台。由于 Xen 采用半虚拟化的技术，也就是说操作系统要经过一点修改才能在 Xen 上运行，所以 Xen 的性能要优于其他虚拟化技术。Xen 由 XenSource 公司负责开发，后来 Citrix 公司在 2007 年 8 月以 5 亿美元收购了 XenSource 公司。IBM、SUN、Intel、HP 等硬件厂商都在自己的硬件上对 Xen 提供了很好的支持。RedHat、Novell 等 Linux 操作系统厂商也都在自己的操作系统上集成了开源的 Xen 软件。

下面将对 VMware 和微软的产品进行介绍。

4. 服务器虚拟化主要产品 VMware

VMware 是服务器虚拟化领域的主要供应商，其最新的产品 vSphere 套件包括许多产品，其架构如图 7-6 所示。

(1) VMware ESX

VMware ESX 是 VMware vSphere 的构造块，ESX 直接安装在为虚拟基础架构提供资源的各个主机服务器的硬件或“裸机”上。ESX 提供了一个稳固的虚拟化层，从而使每个服务器能够容纳多个安全、可移植的虚拟机，这些虚拟机可在同一物理服务器上并行运行。

裸机结构使 ESX 能够完全控制分配给各个虚拟机的服务器资源，并可提供接近本机水



图 7-6　服务器虚拟化示意图

平的虚拟机性能以及企业级的可扩展性。

单个 ESX 最多可以容纳 320 个运行中的虚拟机。假设处于典型工作负载下，每个主机处理器通常支持大约 20 个运行中的虚拟机。使用 VMware Virtual Symmetric Multi-Processing(SMP)时，可以将每个虚拟机配置为最多访问 255 GB 内存和 8 个处理器。在多个虚拟机之间共享物理服务器资源可大大提高硬件的利用率并降低资金成本。

ESX 可提供细致入微的资源管理，通过它可以在运行中的虚拟机之间共享物理服务器的资源，以使服务器利用率最大化，同时确保虚拟机相互隔离。虚拟化起到了资源倍增器的作用，可以让具有 32GB 内存的 4 路服务器从存储区域网络引导 32 个虚拟机，这样就总共具有 64GB 内存、32 个虚拟磁盘和 64 个虚拟网卡。

实际的情况是，有时候没有工作负载，不同的应用程序受制于不同的硬件资源(即有些应用程序受制于内存，而有些应用程序则受制于 CPU)，而且不同工作负载的利用率峰值发生在不同的时间。IT 经理可以根据这些实际情况来合理配置资源。可以使用最小值、最大值和按比例份额数量来为虚拟机分配 CPU、内存、磁盘和网络带宽等资源，这样，应用程序就可以安全并间歇性地使用更多数量的物理资源，而不需要固定的分配额。如果将 ESX 与 vCenter 一起部署，就可以实现对企业数据中心的管理。虚拟机内置了高可用性、资源管理和安全性等特性，这些特性为软件应用程序提供了比静态物理环境更高的服务级别。

VMware vSphere 可以运行在各种经认证的硬件上，从具有多个双核处理器和高端光纤通道 SAN 存储阵列的最大 x86 数据中心系统，到采用低成本的 NAS 和 iSCSI 存储的入门级白盒服务器。

（2）VMware Virtual SMP

VMware Virtual SMP 提供了多处理器虚拟机以处理过重的工作负载。借助 VMware Virtual SMP，单个虚拟机可以同时使用主机服务器中的多个物理处理器或 CPU，从而增强了虚拟机的性能。Virtual SMP 可协助调度非闲置的虚拟处理器，同时又允许处理器过载。通过在虚拟机内部运行的客户操作系统，可取消对闲置虚拟处理器的调度，然后将其重新应用于其他任务。Virtual SMP 会定期在可用的处理器之间移动正在处理的任务，以重新平衡工作负载。VMware 还提供了一项独特的功能，即 Virtual SMP 支持大多数处理器密集型企业应用程序（如数据库、ERP 和 CRM）的虚拟化。

（3）VMware VMFS

VMware VMFS 支持新型分布式服务。虚拟机完全封装在虚拟磁盘文件中，这些文件既可以存储在 ESX 本地，也可以集中存储在共享的 SAN、NAS 或 iSCSI 存储中，如图 7-7 所示。集中存储方式在企业环境中更为常见，这样，其他 ESX 也可以使用共享的 SAN、NAS 或 iSCSI 存储以及 Virtual Machine File System（VMFS）来集中访问各个虚拟机。这种配置的功能要强大得多，因为它允许资源池中包含的多个 ESX 并行访问若干相同的文件来引导和运行虚拟机，并能够对虚拟机存储进行有效的虚拟化。

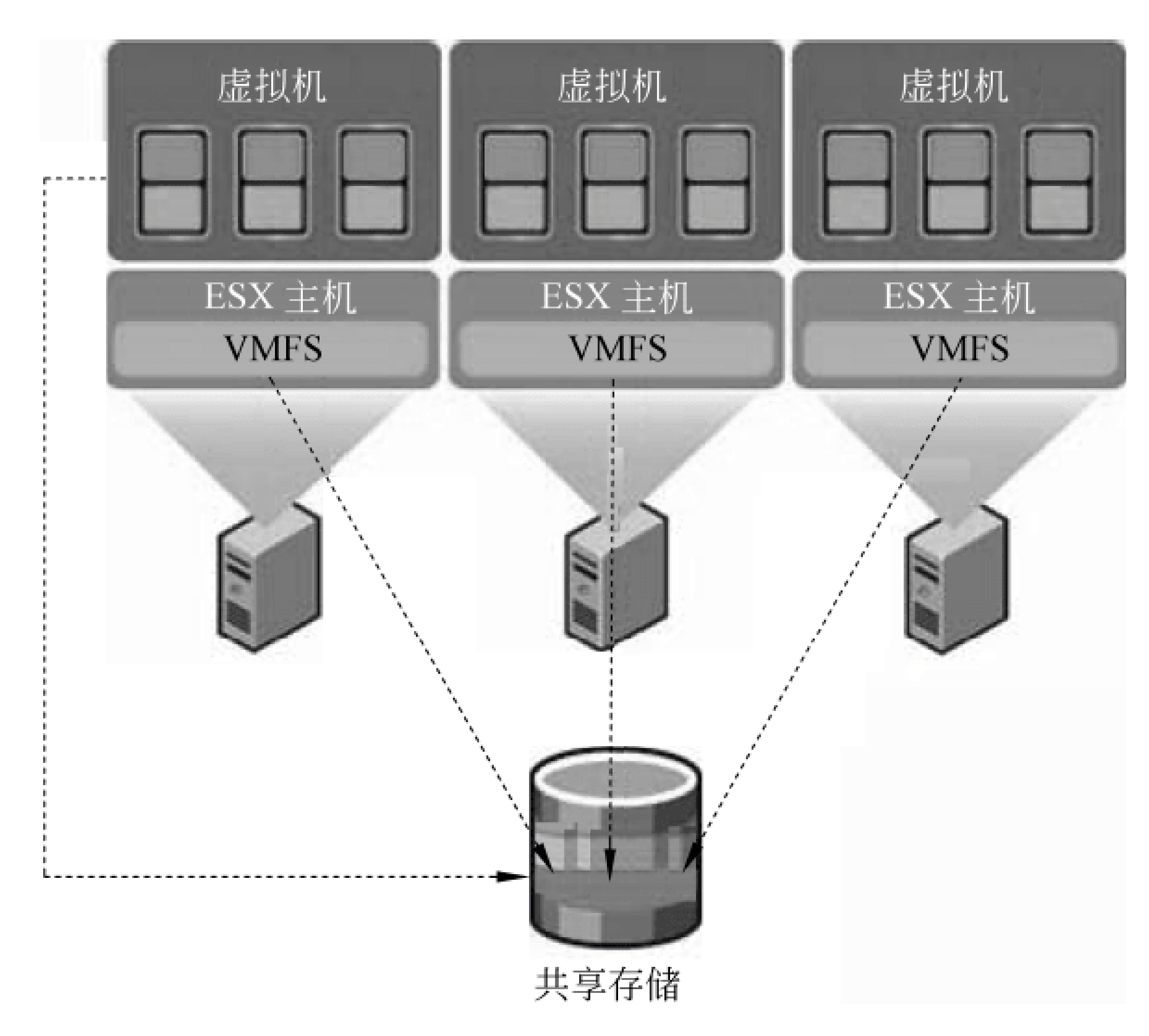


图 7-7　VMware VMFS 分布式服务示意图

常规文件系统只允许一台服务器在指定的时间读写文件系统，而 VMware VMFS 是一种高性能的群集文件系统，它允许多个 ESX 同时对同一个虚拟机存储进行读写。VMFS 提供了磁盘锁定功能，以避免多个服务器同时启动同一个虚拟机。假如某个服务器出现故障，该服务器针对各个虚拟机的磁盘锁将会解除，这样便可以在其他物理服务器上重新启动这些虚拟机。群集文件系统支持一些基于虚拟化的、独特的新型分布式服务。这些服务包括

在两个物理服务器之间实时迁移运行中的虚拟机，在其他物理服务器上自动重启发生了故障的虚拟机，以及跨多个不同物理服务器建立虚拟机群集。由于所有虚拟机均将其存储视为本地连接的 SCSI 磁盘，因此如果将虚拟机迁移到其他物理服务器上，并不需要对虚拟机存储配置进行任何更改。

（4）VMware vCenter

VMware vCenter 用于管理所有 VMware vSphere。vCenter Server 可以集中管理数百个 ESX 主机以及数千个虚拟机，如图 7-8 所示，使 IT 环境具备了操作自动化、资源优化以及高可用性等优势。vCenter 提供了单个 Windows 管理客户端来管理所有任务，该客户端称为 vSphere Client。通过键盘和鼠标可置备、配置、启动、停止、删除、重新定位和远程访问虚拟机。vSphere Client 也可以与 Web 浏览器结合使用，以便通过任意联网设备访问虚拟机。浏览器形式的客户端使用户可以像发送书签 URL 一样轻松地访问虚拟机。

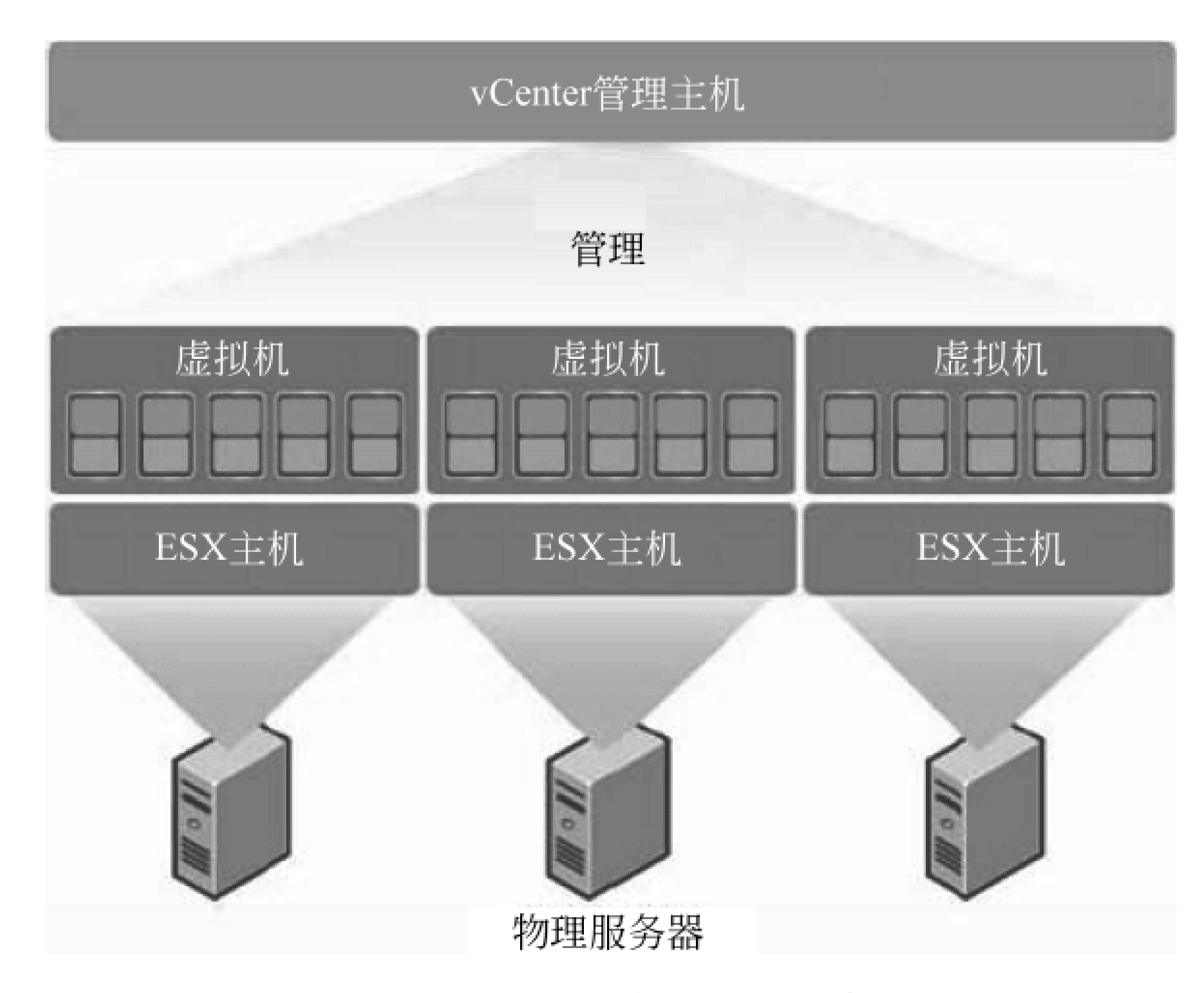


图 7-8　vCenter 管理服务示意图

无论管理多大规模的虚拟化 IT 环境，vCenter 都可以实现最简便、最高效、最安全、最可靠的管理。vCenter 的主要功能包括以下方面。

① 集中管理功能。使管理员能够通过单一界面来组织、监控和配置整个环境，从而降低运营成本。vCenter 提供了多个组织结构分层视图以及拓扑视图，清楚地表明了主机与虚拟机的关系。

② 性能监控功能。包括 CPU、内存、磁盘 I/O 和网络 I/O 的利用率图表，可提供必要的详细信息，用于分析主机服务器和虚拟机的性能。

③ 操作自动化。通过任务调度和警报等功能提高了对业务需求的响应能力，并确保优先执行最紧急的操作。

④ 利用部署向导和虚拟机模板进行的快速置备，大幅缩减了创建和部署虚拟机所需的时间和精力，只需单击几下鼠标就可以完成操作。

⑤ 安全的访问控制机制、强大的权限管理机制以及与 Microsoft Active Directory 的集成，可确保只能对 VMware vSphere 及其虚拟机进行经过授权的访问。通过为经过授权的

管理员和最终用户指派可自定义的角色和权限，可以安全地限制对虚拟机的访问。无论数据中心的访问控制策略多么详尽，也能完全遵守。此外，vCenter 还包括全面的审核跟踪功能，用于保留数据中心内每一项重要更改或操作的详细记录，以便支持新的政府法规，如 Sarbanes-Oxley。

⑥ 编程接口。VMware vSphere SDK 提供了 Web Services API，以便可以通过图形用户界面访问提供的功能和数据，并可以集成第三方系统管理产品以及对核心功能进行自定义扩展。

VMware vCenter 支持将 ESX 主机及其虚拟机组织到群集和资源池中，如图 7-9 所示，这样就大大简化了资源管理工作。群集是虚拟基础架构管理中的一个新概念，它同时具有多个主机服务器的强大功能与管理单个实体的便利性。利用资源池功能和内在高可用性，群集可将多个独立的主机聚集到单个群集中，从而大大简化了服务器的管理工作。现在可以将虚拟机置备到群集中而不是单个 ESX 主机上，这样虚拟机便可使用群集中的所有资源。vCenter 可以为虚拟机选择最适合的主机，并可以在情况发生变化时在群集内部移动虚拟机。

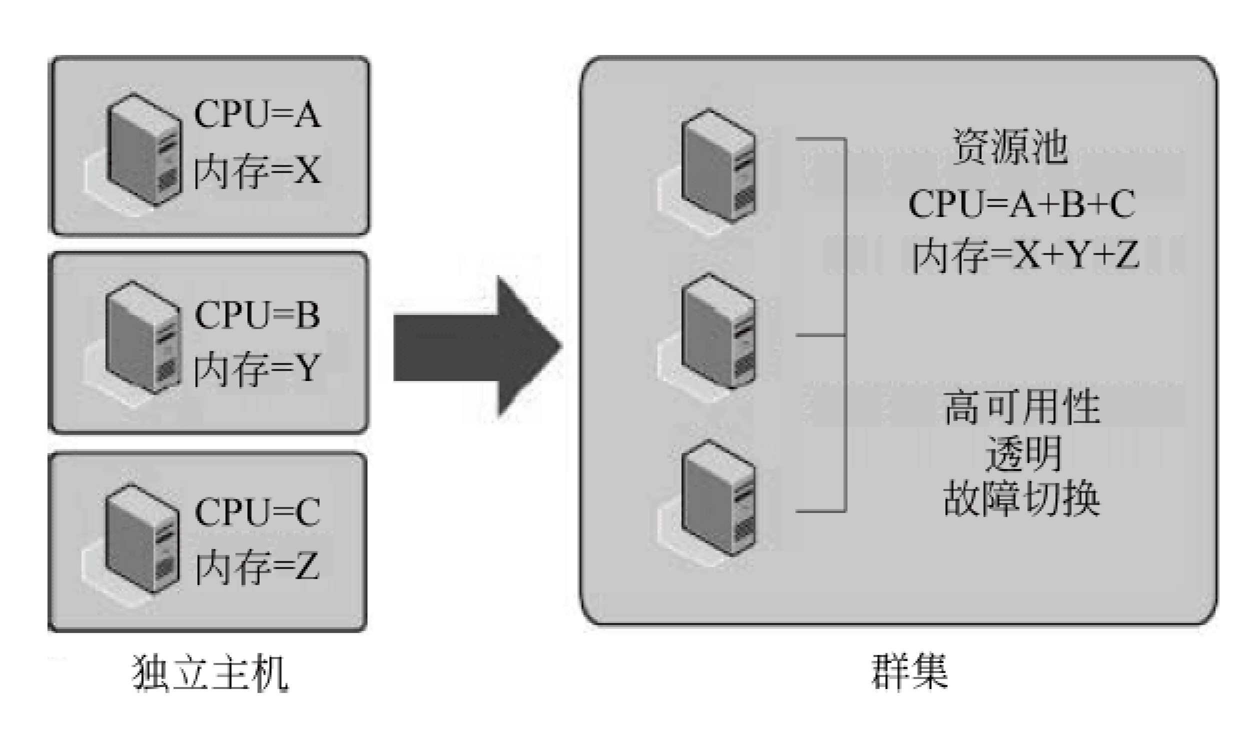


图 7-9　虚拟化集群示意图

由于虚拟机现在是运行在群集上而不是独立的 ESX 主机上，因此 VMware 群集具有内在的高可用性。如果某个 VMware 主机出现故障，则可以在群集中的其他主机上重新启动该主机上的虚拟机。当在群集中添加或删除了主机时，群集中的虚拟机可使用的资源就会随之动态地增多或减少。

资源池通过将独立主机或群集的资源细分到更小的池中，进一步简化了虚拟基础架构的管理工作并提高了灵活性。资源池是用来容纳虚拟机的容器，配置有一组 CPU 和内存资源，供该资源池中运行的虚拟机共享。资源池的一般用法是，将对一组精确指定的资源的控制权指派给一组或一个用户，但不授予他们对底层物理环境的访问权。

资源池是一种理想的解决方案，适合用来为用户授予创建和管理其虚拟机的权限，同时限制他们对资源的使用。例如，可以为需要管理虚拟机的开发小组提供一个如图 7-10 所示的资源池，该资源池共有时钟频率为 12GHz 的 CPU 和容量为 12GB 的内存。然后，开发小组可以创建和控制自己的虚拟机，但无论启动多少个虚拟机，资源消耗量绝不会超过资源池的容量。资源池可以进一步细分，可以将 12GHz 的大型开发资源池进一步划分成更小的资源池，供各开发人员单独使用。这样，资源池就简化了虚拟基础架构的管理，无须在置备虚

拟机时单独为虚拟机预先配置资源分配额。为充分利用共享的虚拟基础架构,可以对资源池进行配置,允许它们在活动高峰期“爆发”,以使用群集上邻近资源池中任何可用的浮动容量甚至闲置资源。

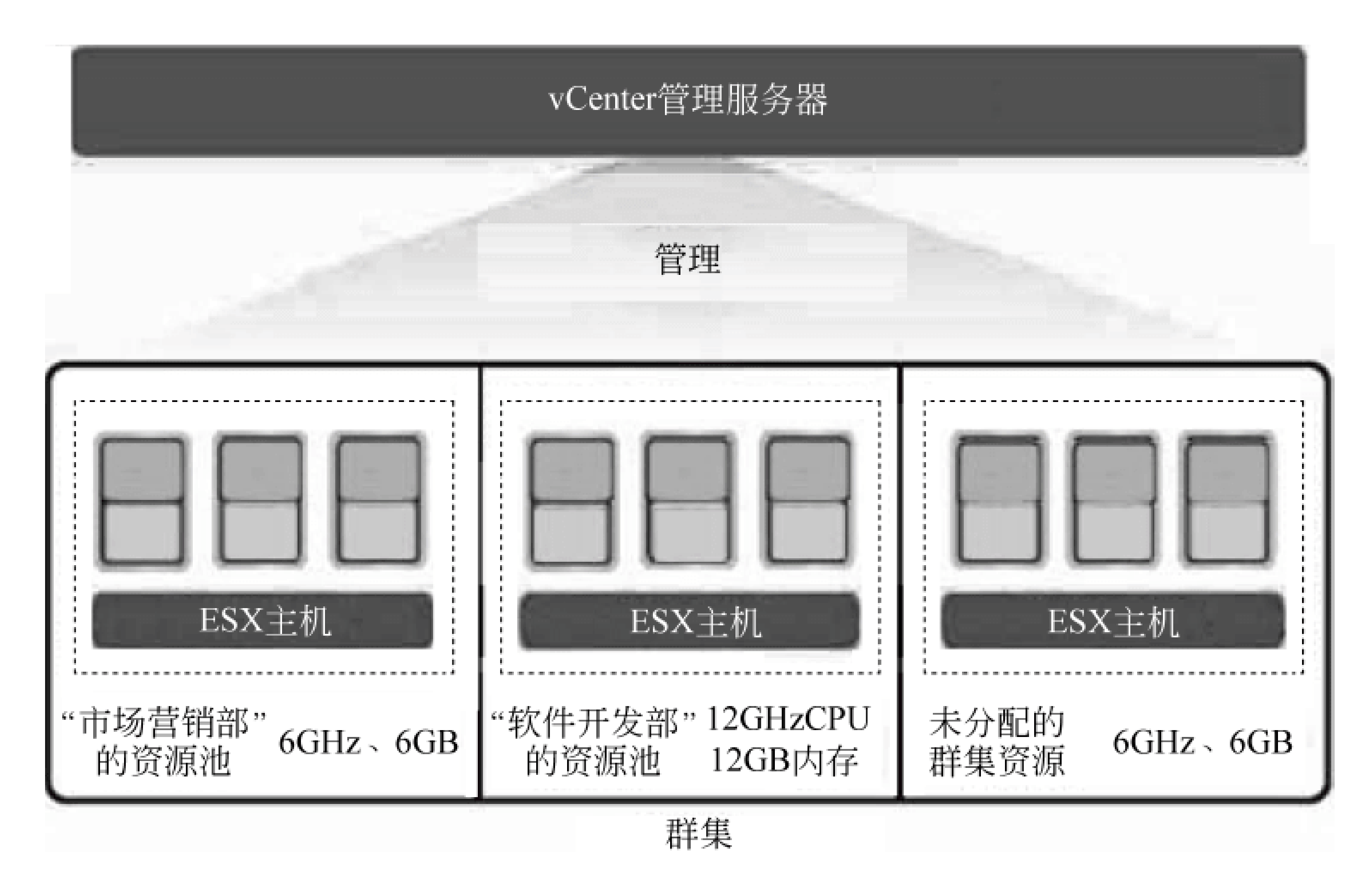


图 7-10　资源池资源分配示意图

资源池的资源分配也可以动态变更,这一特性对工作负载不断发生波动的企业应用程序来说非常有利。例如,可以将一个多层 SAP 安装包配置为单个资源池中的若干联网虚拟机。如果预计将出现 SAP 活动高峰期,系统管理员只需为 SAP 资源池分配更多的 CPU 和内存即可,而不必逐个调整各个 SAP 虚拟机的资源分配。资源池灵活的分层结构使用户能够在业务部门之间轻松协调可用的 IT 资源。各业务部门可以采用专用基础架构,同时仍然能够受益于资源池的高效性。

(5) VMware VMotion

VMware VMotion 支持虚拟机在主机之间的实时迁移。作为动态、自动化及自我优化的数据中心的一个关键启动组件,VMware VMotion 支持在物理服务器之间实时迁移运行中的虚拟机,同时又可以避免宕机,确保连续的服务供应以及处理过程的完整性。借助虚拟机实时迁移技术,公司在执行硬件维护时无须安排宕机和中断业务操作。VMotion 还可以使资源池内的虚拟机持续进行自动优化,最大限度地提高硬件的利用率、灵活性和可用性。使用 VMotion 在物理服务器之间实时迁移虚拟机是通过三项基础技术实现的。

首先,虚拟机的整个状况封装在共享存储区(如光纤通道、iSCSI 存储区域网络(SAN)或网络连接存储(NAS))上的一组文件中。VMware 的群集虚拟机文件系统(VMFS)允许多个 ESX 并行访问同一组虚拟机文件。

其次,虚拟机的内存映像和精确的执行状况可通过高速网络在各 ESX 主机之间迅速传递。VMotion 通过在一个位图中保持对现行内存处理过程的跟踪,使用户在传递期间察觉不到性能变化。一旦整个内存和系统状况被复制到目标 ESX 后,VMotion 就会中止源虚拟机的运行,将位图复制到目标 ESX,并在目标 ESX 上继续运行该虚拟机。整个过程在千兆位以太网上只需要不到两秒钟的时间。

再次,虚拟机使用的网络也被底层 ESX 虚拟化,确保即使在迁移之后,虚拟机的网络身

份标识和网络连接也能保留下来。VMotion 会在此过程中管理虚拟 MAC 地址。一旦目标虚拟机被激活，VMotion 就会对网络路由器执行 ping 指令，以确保它知道该虚拟 MAC 地址的新物理位置，如图 7-11 所示。由于使用 VMotion 进行虚拟机迁移可保留精确的执行状况、网络身份标识和活动的网络连接，因此可以实现零宕机，不会对用户造成干扰。



图 7-11　VMotion 技术示意图

（6）VMware DRS

VMware DRS(Distributed Resource Scheduler)可达到 80％的利用率，同时能够保证较高的服务级别。VMware DRS(Distributed Resource Scheduler)与 VMware vSphere 配合使用，可以在虚拟基础架构中不断自动平衡同一群集中各虚拟机的工作负载。在群集中首次启动某个虚拟机时，VMware DRS 会自动找出具有足够资源的 ESX 主机来运行该虚拟机。如果所选主机的情况发生变化，例如，其他虚拟机的活动增加，使该虚拟机无法实现最低资源分配保障，VMware DRS 将会发现这一情况，并在群集上搜索能够满足该虚拟机资源分配需求的备用 ESX 主机。然后 VMware DRS 会使用 VMotion 自动将虚拟机迁移到新主机上，用户操作和应用程序均不会受到任何影响。这样，在虚拟基础架构中，所有服务器工作负载便可实现持续平衡。VMware DRS 通过 ESX Local Scheduler 和 vCenter Global Scheduler 来进行操作。ESX Local Scheduler 可根据当前的工作负载来决定将主机中的哪些处理器用于虚拟机的执行，只要发现其他的主机处理器能够提供更多容量，便会重新分配虚拟机，也许每隔几毫秒便会重新分配一次。与此不同，vCenter Global Scheduler 则会在 ESX 主机所在的整个群集内持续评估放置虚拟机的最佳位置。

Global Scheduler 会决定由哪个 ESX 容纳新启动的虚拟机。如果其他 ESX 主机能够提供更适合的资源集，Global Scheduler 就会使用 DRS 重新分配虚拟机。

VMware DRS 可以配置为以自动或手动模式运行。在自动模式中，VMware DRS 会将虚拟机迁移到群集中最适合的主机上，无须进行任何干预。在手动模式中，VMware DRS 会就虚拟机的最佳位置提出建议，然后让系统管理员决定是否进行更改。借助 VMware DRS，可以将新的虚拟机放置到群集上，而不是特定的主机服务器上。对于放置的位置以及启动的时间，VMware DRS 会做出明智的决定。VMware DRS 还支持在特定使用情况下应用关联性和反关联性规则。例如，反关联性规则可使群集中各虚拟机始终在不同的物理服务器上运行，以便实现硬件冗余。相反，关联性规则可使两个具有内部联网要求的虚拟机始终位于同一物理主机上。迁移虚拟机后，VMware DRS 将会保留全部已分配的资源。该组

件能认识到：如果在具有 3GHz 处理器的八路主机上，虚拟机分配到 10%的 CPU 资源，则迁移到处理器速度较慢的双路主机上以后，该虚拟机就需要获得更高比例的主机资源。在群集中添加新的 ESX 主机时（这在 vCenter 内只是一个简单的拖放操作），VMware DRS 将立即进行响应。新的主机将会扩展群集中各虚拟机可以使用的资源池，而 VMware DRS 会适当地将虚拟机迁移到新的主机上，以重新平衡工作负载。同样，从群集中删除主机时，VMware DRS 也会进行响应，将该主机上的虚拟机迁移到群集中的其他主机上。使用 VMware DRS 的最终结果是，数据中心能够以 80%以上的利用率水平可靠地运行，同时可以保障所有应用程序的服务级别。利用 VMware DRS，只需进行最少的容量规划工作，便可从 x86 服务器的投资中获取更高的回报率。

（7）VMware HA

VMware HA（High Availability）为虚拟机中运行的应用程序提供了易于使用、经济高效的高可用性功能。由硬件故障所导致的 ESX 主机的缺失不再是灾难性的事件，而只是意味着群集可以使用的资源池缩减了。在这种情况下，VMware HA 会在群集中的其他 ESX 主机上为故障主机上的虚拟机重新分配资源并重新启动这些虚拟机，vCenter Global Scheduler 则会决定放置这些虚拟机的最佳位置以满足资源需求。

通常可以借助故障切换群集产品（如 Microsoft Cluster Services 或 Veritas Cluster Services）来实现应用程序的高可用性，但这些产品不仅价格昂贵，而且难以配置和管理。故障切换群集需要企业支付不菲的费用来升级操作系统或购买第三方软件，并且所保护的应用程序还必须支持群集。

故障切换群集还会消耗大量资源，因为备用群集节点需要独占硬件，即便它们未处于活动状态也是如此，如图 7-12 所示。

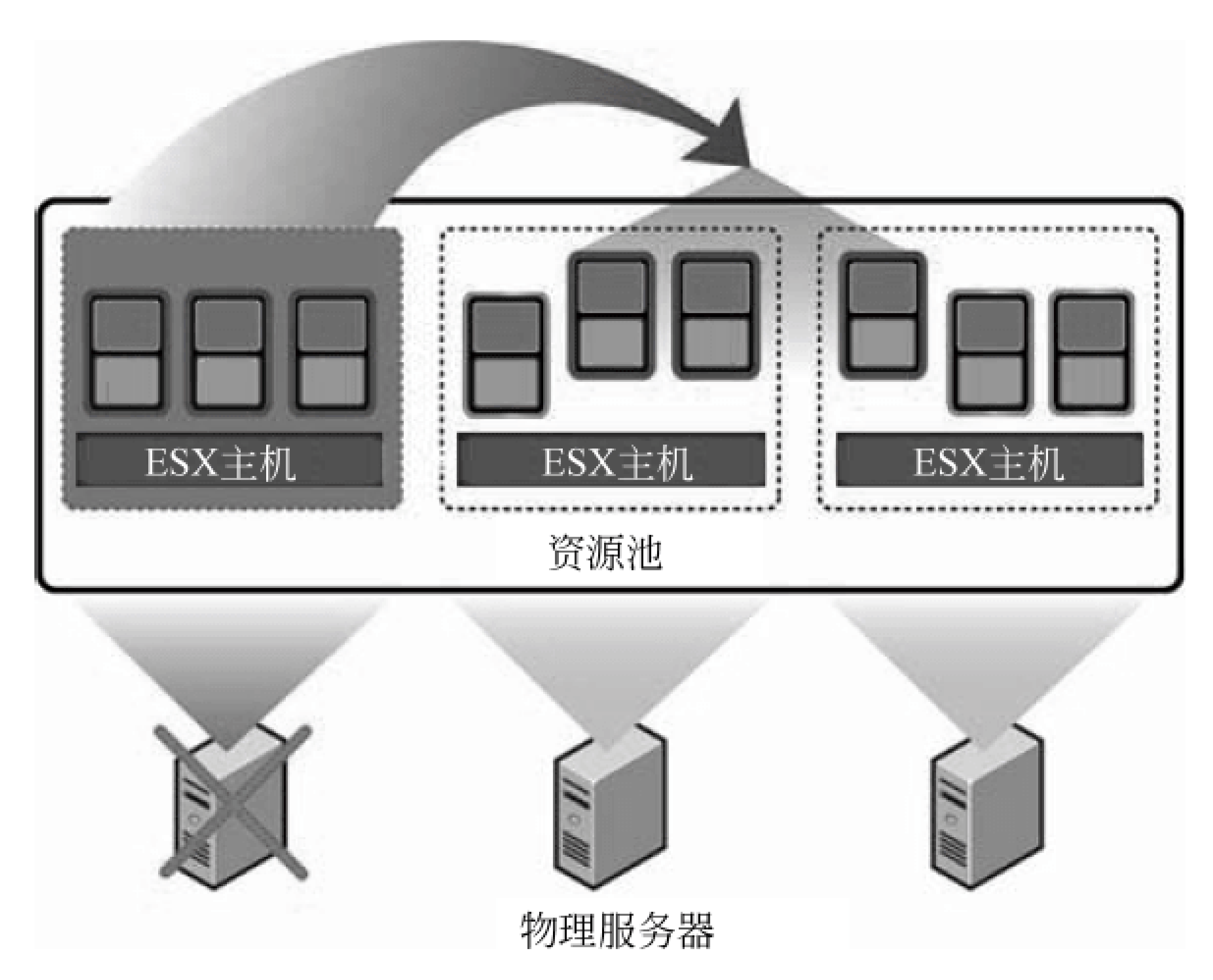


图 7-12　VMware HA 技术示意图

VMware HA 无须进行任何配置即可提供高可用性。只要为群集或主机选择 VMware HA 选项，其所有虚拟机均会得到保护，使虚拟机在主机发生故障之后可以自动重新启动。VMware HA 与故障切换群集的不同之处在于，重新启动虚拟机时会有一小段宕机时间，但对于大多数应用程序而言，极其短暂的中断是可以接受的，而且 VMware HA 可以避免故

障切换群集所引起的费用和复杂性。需要注意的一点是，在受 VMware HA 保护的群集中，vCenter 管理主机不会发生单点故障。在每台服务器上安装的 VMware HA 代理会不断向资源池中的其他服务器发出“心跳”信号，一旦“心跳”信号丢失，所有受影响的虚拟机都会立即在其他服务器上重新启动。正是由于 VMFS 群集文件系统允许多个 ESX 拥有对相同虚拟机文件的读写权限，才使虚拟机的重新启动得以实现。VMware HA 可确保资源池中始终有充足的资源，以便当某个服务器出现故障时，能够在其他物理服务器上重新启动虚拟机。

(8) VMware Consolidated Backup

VMware Consolidated Backup 提供了不经局域网的备份功能(LAN Free Backup)，实现了零宕机，如图 7-13 所示。VMware Consolidated Backup 提供了一个易于使用的集中式工具来执行不经局域网的备份操作，这种操作可保留文件级别的可视性。VMware Consolidated Backup 将在停止虚拟机中的应用程序后为运行中的虚拟机创建快照，从而在磁盘中集中处理备份工作，以确保文件系统的一致性。然后，由一个 Windows 备份代理服务器来装载这些虚拟磁盘快照，该服务器可以使用标准的备份代理程序将备份存储到磁带或磁盘设备中。VMware Consolidated Backup 预先集成了常用的备份实用程序，并且提供了预处理和后处理脚本，无须任何额外准备工作便可轻松实施。VMware Consolidated Backup 将透明地运行，而不需要中断虚拟机的活动。备份处理过程在 ESX 主机外部进行，因此不会对虚拟机中重要应用程序所需的 CPU 和网络资源造成影响。备份期间，不会发生系统中断，也不会影响到生产服务器。由于 Consolidated Backup 只需在代理服务器(而不是每个虚拟机)上运行一个备份代理程序，因此减少了所需的备份代理程序许可证数量，并提高了可管理性。使用代理服务器还可以降低 ESX 的负载，使 ESX 可以在同一物理服务器上运行更多的虚拟机。基于文件的完整增量备份在运行 Microsoft Windows 操作系统的虚拟机上受到支持；而针对灾难恢复方案的完整映像备份则适用于所有虚拟机，无论虚拟机的客户操作系统是什么。

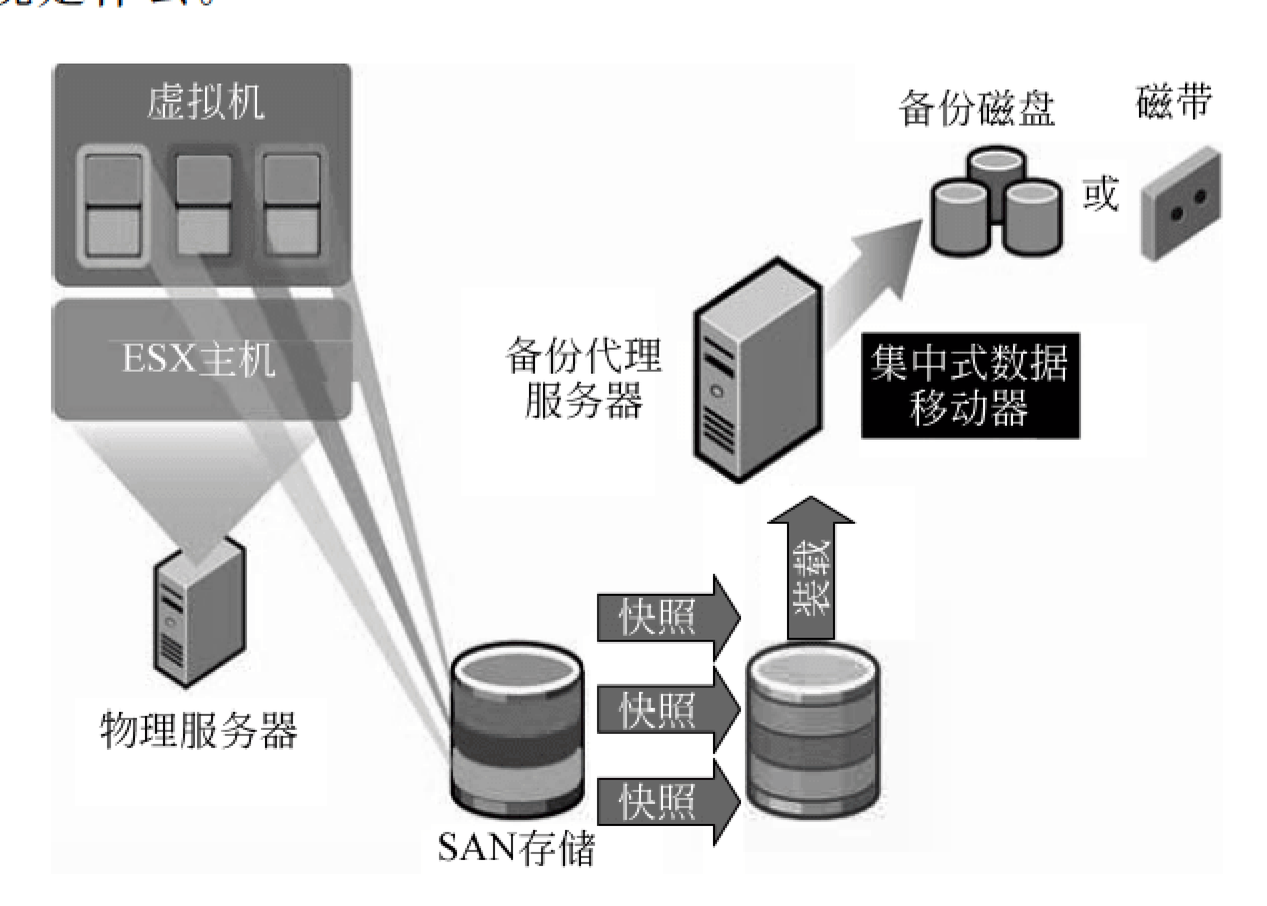


图 7-13　VMware Consolidated Backup 备份技术示意图

下面介绍 VMware 虚拟化软件的优势。

(1) VMware vSphere 可以降低成本

全球有 20 000 多家 IT 企业采用了 VMware 虚拟化软件，该软件为这些企业节约了

数十亿美元的硬件成本和运营成本。随着处理器功能的日益强大、VMware vSphere 在更多低成本硬件上得到认证，以及该套件不断扩展以适应最苛刻的企业工作负载条件，由 VMware vSphere 所节约的成本还会不断提高。

（2）实现服务器的整合与控制，同时使服务器利用率最大化

通过在数量更少、可高度扩展并可靠的企业级服务器上的虚拟机中运行软件应用程序，控制服务器数量的膨胀。VMware vSphere 的客户通常能够在每个物理处理器上整合10 个或更多个虚拟机，因此大大提高了服务器利用率，并有效控制了服务器数量的膨胀。

通过整合未充分利用的服务器，就不再需要为新项目另外购置硬件，从而减少资金投入成本。通过从数据中心删除服务器，就可以在供电、制冷和场地方面减少一定比例的运营成本。由于一个虚拟机目前最多可以支持 4 个处理器和 16GB 的内存，因此企业当前 95％的工作负载可以实现虚拟化和整合。

（3）在企业范围内实现标准化

由于虚拟化是将软件从硬件中抽离出来以创建可移植的虚拟机，因此利用 VMware vSphere 可以在企业范围内更轻松地实现数据中心的标准化。VMware vSphere 可以在所有主要硬件供应商的塔式、机架和刀片式服务器上运行大多数主流操作系统，这大大提高了现有的多供应商硬件投资的价值。

（4）简化 IT 操作，提高管理效率

VMware vSphere 可以简化人工和资源密集型 IT 操作，例如跨越不同的硬件、操作系统和软件应用程序环境进行服务器的置备与维护，使更少的 IT 员工能够承担更多的工作负载。

此外，vCenter 提供的统一监控和管理平台可显著提高系统管理员的工作效率，使每个系统管理员能够监控并有效管理大型基础架构资源池。

（5）减少业务部门的 IT 协调成本

各个业务部门的协调工作也许是最为棘手，也最容易被忽视的工作之一，它耗费了 IT 员工的大量时间。但 IT 企业采用 VMware vSphere 之后，由于硬件管理与软件管理分开了，因此部门协调的成本可大幅减少。例如，在采用 VMware vSphere 之前，IT 员工需要花费大量的时间，针对业务部门的应用程序来协商硬件宕机时间。现在，由于可以将运行中的软件从需要维护的硬件上移走，而不会对业务部门产生影响，因此可以随时执行硬件宕机，从而节约整体 IT 管理成本。

（6）简化软件开发与测试

VMware vSphere 在许多方面简化了软件的开发与测试。大幅简化了一般的耗时性工作，如配置服务器、置备服务器以及将配置存档和还原，从而提高了开发人员的工作效率。此外，使用虚拟机将开发、测试和转储环境整合到共享硬件后，所需的硬件比原来要少得多。最后，由于使用了虚拟机，可以更轻松地扩大测试范围并提高软件质量。

（7）VMware vSphere 简化了管理

VMware vSphere 可将所有基于 x86 的操作系统的管理统一到跨越数据中心的单个虚拟硬件平台上。它使置备、取消置备和回滚的速度达到了实时级别。另外，它还强制基于模板（而不是策略）来部署服务器，与手动创建服务器相比，管理员使用“黄金”主模板后节省了

大量时间。由于虚拟基础架构是统一的，因此服务器的部署过程也是一致的，这使运营风险大大降低了。

(8) 安全集中的数据中心管理

VMware vSphere 简化了监控、管理、报告和远程访问等操作，用户可以从数据中心内的任意位置通过 vSphere Client 来执行这些操作。如果要对系统软件和配置执行操作，则无须访问服务器。浏览器形式的客户端使用户可以像发送书签 URL 一样轻松地访问虚拟机。

此外，VMware vSphere 提供了跨 Windows、Linux 和 NetWare 服务器的统一管理平台。现在，管理员只需学习一种置备和监控系统的方法，而不必每部署一个操作系统版本就学习一种相应的方法，这样可以降低培训成本，并且使各个操作系统应用的策略更加一致。

(9) 确保服务器构建过程的一致性

vCenter 基于模板来置备服务器。由于基于预先配置的操作系统和应用程序组合来进行置备，因此 IT 管理员可以确保环境中运行的所有服务器在安全性和配置方面均符合当前的最佳做法。因此，各个 Exchange Server 彼此非常相似。

由不同管理员构建的服务器彼此也是相似的。这意味着故障排除更加简单，发生端口意外开放或残留漏洞服务的可能性几乎降低为零。基础架构从根本上实现了合理化，避免了反常的差异。

(10) 提高修补程序大范围应用的成功率

既然基础架构实现了合理化，虚拟基础架构中内置的服务器构建过程一致，那么 IT 管理员就可以确信，如果某个修补程序没有中断一种类型的服务器，它就不会中断其他类型的服务器。并且可以在测试用的服务器中创建与当前生产系统完全相同的系统副本，以进行修补程序和升级测试。所创建的系统副本与还原的备份或磁盘映像不同，因为虚拟机副本与源系统完全相同(包括虚拟硬件层)。此外，借助虚拟基础架构中提供的快照和回滚功能，虚拟机在安装修补程序后如果出现故障而无法运行，则可以立即回滚到前一个已知的正常状况。

(11) 部署虚拟工具可简化更改管理

虚拟工具是经过完整预配置的虚拟机，包含操作系统和软件应用程序。虚拟工具结合了软件部署的简便性与预配置设备所具有的优势，正在逐渐使软件分发模式发生转变。对解决方案提供商来说，构建虚拟工具比构建硬件工具更简单、更经济。防火墙恰恰反映了这种模式转变。最初的网络防火墙是软件程序。要安装防火墙，用户就必须购买计算机，安装支持的操作系统，然后安装防火墙并配置所有功能。为了降低防火墙部署的复杂性，供应商构建了基于硬件的防火墙。这些防火墙工具可以是标准硬件，也可以是定制硬件，其中包含最低配置的操作系统和完整的防火墙程序。针对该问题的另一种更新的解决方法是：防火墙虚拟工具。使用这种方法时，预配置的防火墙会安装在虚拟机中，并且可以部署到现有的硬件上。

(12) 简化了旧版软件的迁移

虚拟基础架构允许根据需要在更高版本的硬件上使用虚拟机继续运行旧版应用程序(需要旧版操作系统)。这一特性经证实有助于公司延长旧版软件资产的生命周期，增加了

其使用价值,同时避免了高额的移植成本。在虚拟机中管理旧版系统可以大大提高可靠性,减少维护费用。

(13) VMware vSphere 可提高敏捷度

VMware vSphere 针对 IT 部门提供了一些功能,可显著提高 IT 部门对业务部门需求的响应速度。由于虚拟基础架构切断了软硬件之间的制约关系,因此 IT 公司可以更灵活、快速地置备新服务器和调整资源,以适应业务要求的变化。

(14) 瞬时置备新服务器

无论是一周后需要一个新服务器,还是一小时后需要 50 个服务器,对 VMware vSphere 来说都不是问题,它提供了强大的瞬时置备功能,可以跨 Windows、Linux、Solaris x86 或 NetWare 等多个操作系统实时置备和取消置备服务器。IT 公司可以实施及时的服务器置备方案,使业务部门能够根据需要置备他们自己的服务器。设想一下,当业务部门打电话申请新的服务器时,立即告诉他们新服务器已经启动并开始运行,正在等待他们登录会是怎样的情景。同样,扩展应用程序,甚至执行需要重新引导服务器的日常维护请求等操作,均能以更快的速度完成。

借助 VMware vSphere,管理员可以从标准服务器模板库中快速选择“黄金”模板以进行新服务器部署,并在几秒钟内将模板部署到硬件池中。VMware vSphere 可执行文件复制,以便为所选的服务器模板创建一个新实例,然后对其进行配置以便使用。

服务器部署成为一项如此低成本的操作,以致 IT 部门创建服务器的成本远远低于部署完整物理服务器的成本,例如,创建一个临时服务器来测试 Beta 版应用程序软件成为轻而易举的事。使用 VMware vSphere 置备服务器只需几秒钟,而手动部署服务器则一般需要几个小时或几天时间,两相比较,采用虚拟基础架构所节约的成本就迅速显现出来了。

(15) 为业务部门提供了实用程序计算功能,以保持稳定的服务级别

借助资源池,IT 公司可以对应用程序和工作负载要求的变化迅速做出响应,根据业务优先级来轻松调整计算资源,以确保服务级别。IT 公司可指定用来控制虚拟机资源分配的规则和优先级,VMware vSphere 将会持续自动地优化虚拟机的位置,以便提高硬件的利用率、灵活性和可用性。这样,IT 公司就能为业务部门提供专用的基础架构,同时仍能通过资源池来获得更高的硬件利用率。采用 VMware vSphere 后,只需部署并使用少量的平台即可灵活应对不断变化的需求。

(16) 使所有应用程序均能受益于高端硬件的性能和可靠性的提高

通过 VMware vSphere,可以在数据中心以低成本部署高端硬件,例如具有冗余组件的机架服务器和 4 向刀片式服务器。与低端硬件相比,高端服务器虽然成本较高,但可以承担更多的工作负载,因此,对高端硬件的初始投资会迅速得到回报,具体表现为利用率得到提高以及由于管理的服务器较少而节约了运营成本。此外,每一项工作负载都可以根据需要利用高端硬件的增强功能,从而能够为最终用户提供卓越的应用程序性能和可靠性。

(17) VMware vSphere 可增强安全性

VMware vSphere 提供了一致、安全、可审核的数据中心环境,数据中心可以由非同类硬件构成。虚拟机内运行的操作系统仍然需要安全性管理和漏洞修补程序,但由于

VMware vSphere 强制实行基于角色的精细访问权限控制，因此可以大大提高操作系统的稳定性和安全性，简化访问权限的管理。

(18) 集中保护和审核数据中心

由于通过细致而灵活的访问权限控制来保护基础架构的虚拟机配置和远程访问，因此仅有极少数 IT 员工需要直接访问 VMware vSphere 服务器硬件。管理员和最终用户可以远程执行所有的服务器置备和配置操作，同时，综合审核日志中会记录所有重要的操作。此外，也可以通过管理虚拟机的访问权限来控制用户对自身未提供足够安全性的应用程序的访问。

(19) 在硬件级别隔离故障和安全性

由于各虚拟机在运转期间彼此完全隔离，因此除非通过网络通信，否则运转异常或被危及的应用程序不会影响到同一环境中的其他虚拟机。借助故障隔离功能，经过适当配置的虚拟机可以更好地抵御数字攻击，因为一个虚拟机不会危及其他虚拟机。通过 VMware vSphere 虚拟网络连接，管理员能够灵活地将虚拟机从公司网络中隔离出来，或者使它们与网络中的其他物理机完全对等。

(20) VMware vSphere 可以提高可用性

VMware vSphere 可以实现硬件维护和服务器备份所需的零宕机，从而显著提高应用程序的可用性，使应用程序的正常运行时间接近 100%。此外，VMware vSphere 还可以通过 VMware HA 轻松地使虚拟机以低成本实现高可用性。

(21) 支持零宕机维护

虚拟基础架构最值得关注的一点或许就在于，IT 管理部门在安排员工任务方面具有更大的灵活性。通过将硬件维护任务与软件维护任务分开，因需要宕机而推迟的管理操作大量减少。借助 VMware vSphere，IT 部门只需将 ESX 主机置于维护模式即可，DRS 会自动将所有虚拟机迁移到资源池中的另一个 ESX 上，这样，维护物理服务器时就不需要宕机。维护工作可以在主要工作时段(上午 8 时到下午 5 时)执行，而不需要在夜里和周末安排宕机。同样，可以随时为运行中的生产系统创建快照，以进行调试或修补程序测试。如果新的修补程序或应用程序升级出现问题，可以进行离线检查，而不需要中止服务器来进行维护。这样就能安排最有能力解决问题的员工以最快的速度解决该问题。

(22) 支持零宕机备份：

借助 Consolidated Backup，可以将虚拟机作为虚拟磁盘来备份或者使其具有文件级别的可视性，而不需要宕机，也不会对虚拟机和 LAN 的性能产生任何影响。Consolidated Backup 将在停止客户操作系统的文件系统后创建虚拟机快照，以确保文件的完整性。然后，由一个 Windows 备份代理服务器来装载这些虚拟磁盘快照，该服务器可以使用标准的备份代理程序将备份存储到磁带或磁盘设备中。

(23) 通过简单、迅速的灾难恢复功能提供业务连续性高级保护

传统的高可用性解决方案通常比较复杂和昂贵，一般专门用于关键应用程序。而 VMware vSphere 降低了实现高可用性的经济成本，使至今仍未受保护的大多数软件应用程序可以实现高可用性。借助 VMware HA，公司可以实施一个统一的灾难恢复平台，在发生硬件故障时可以恢复多个生产虚拟机，而不需要投入大量资金来进行生产硬件与灾难恢复硬件的一对一映射。VMware HA 以低成本为虚拟机中运行的所有应用程序提供

了高可用性。与其他专用于特定操作系统或软件应用程序的高可用性解决方案不同，VMware HA 可为整个 IT 环境提供一致、易于管理的高可用性解决方案，作为可靠的“第一道防线”。

(24) 使用 VMware vSphere 构建虚拟化企业

作为唯一可供生产环境使用的虚拟化套件，VMware vSphere 已在 20 000 多家规模不等的企业客户中实施，在各种环境和应用程序中均得到验证。该套件针对各种应用广泛的硬件、操作系统和软件应用程序进行了充分优化和严格测试，并获得了认证，可用来在企业范围内实现标准化，而不管企业的操作系统和硬件如何。VMware vSphere 提供了内置的管理、资源优化、应用程序可用性和操作自动化等功能，不仅可以节约改造成本，而且还可以提高运营效率、灵活性和 IT 服务级别。VMware vSphere 可扩展以支持任何规模的 IT 环境，而不局限于任何操作系统，因此客户能够自由选择所需的操作系统和软件应用程序。

5. 服务器虚拟化主要产品(Microsoft Hyper-V)

在 VMware 公司成立的第 4 年，也就是在 2003 年，微软收购了推出了 Virtual PC 软件的 Connectix 公司，并在其后推出了服务器虚拟化(Virtual Server)软件，如图 7-14 所示，开始了微软的虚拟化之路。

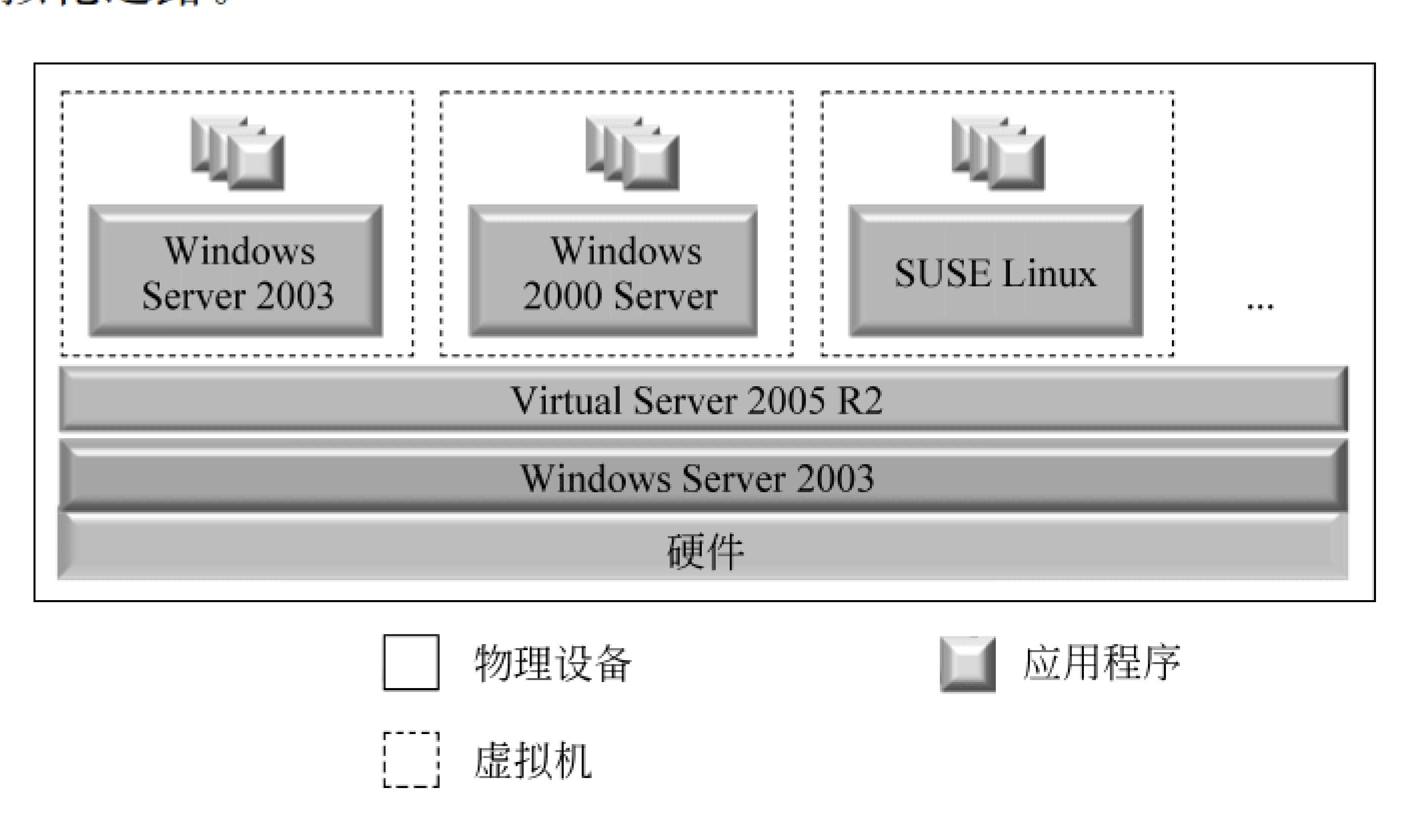


图 7-14　Virtual Server 虚拟化架构示意图

在 2008 年，推出了最新的虚拟化产品 Hyper-V，如图 7-15 所示。Hyper-V 是一个 Hypervisor(系统管理程序)，开发代号为 Viridian，它的主要作用就是管理、调度虚拟机的创建和运行，并提供硬件资源的虚拟化。Hyper-V 是微软伴随 Windows Server 2008 最新推出的服务器虚拟化解决方案，集成在 Windows Server 2008 中。

Hyper-V 的本质是一个 VMM(虚拟化管理程序)，与微软之前的 Virtual Server 系列产品处在的层次不同，它更接近于硬件，这一点比较像 VMware 的 ESX Server 系列，实际上 Hyper-V 属于微软的第一个裸金属虚拟化产品(Bare-Metal Virtualization)。其架构如图 7-15 所示。

(1) Hyper-V 的功能

Hyper-V 的功能如下：

- 支持 32 位(×86) & 64 位(×64) VMs；

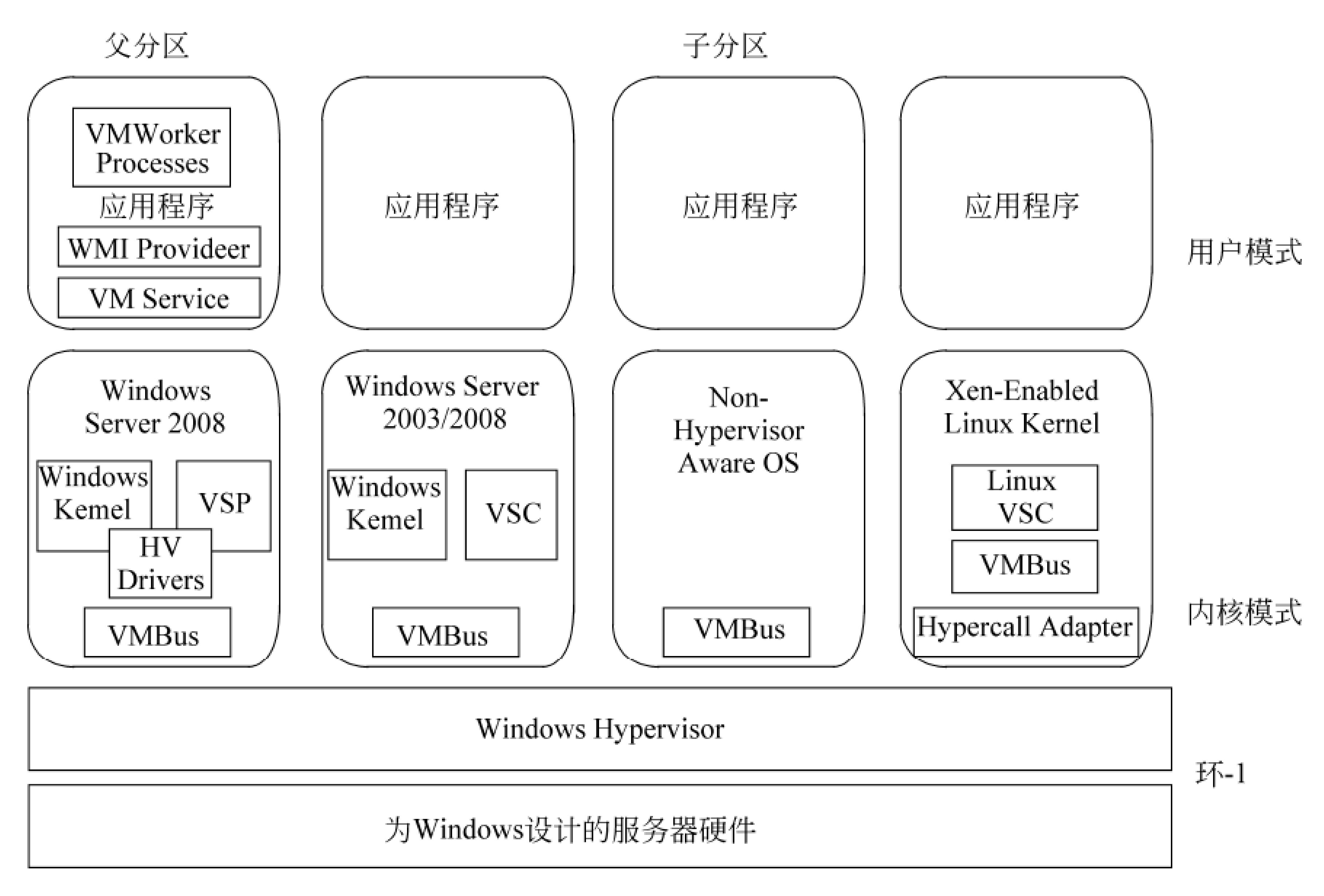


图 7-15　Hyper-V 架构示意图

- 支持大内存(每 VM 为 64GB);
- 支持多 CPU(最多 4 核);
- 支持群集(Clustering)的快转移,有高可用性;
- 支持 Volume Shadow Service;
- 支持虚拟机"直达"(Pass-through)硬盘;
- 支持虚拟机快照(Snapshots);
- 支持最新的硬件架构(VSP/VSC/VMBus)Disk、Networking、Input、Video;
- 有完善的网络服务 VLANs 和 NLB;
- 有 DMTF 标准的 WMI 管理界面;
- 支持 Server Core 和全 Windows 安装。

(2) Hyper-V 的优点

Hyper-V 提供先进的裸金属虚拟化技术,优点如下。

① 对 64 位高性能体系的支持。全新的 64 位微内核 Hypervisor 架构使 Hyper-V 可以提供更广泛的设备支持,如对大容量内存的支持等,以及增强的性能和提升安全性,能够承载更多的虚拟机运行实例。

② 支持广泛的操作系统。为了更好地满足企业的 IT 现有环境及未来的 IT 发展趋势,Hyper-V 广泛支持在虚拟化环境中同时运行同类型的操作系统,包括 32 位和 64 位的多种不同服务器平台操作系统,例如 Windows、Linux 等操作系统。

③ 支持对称多处理器。面对当今以对称多处理器(SMP)为主流的服务器,Hyper-V 可在一个虚拟机环境中最多支持四个虚拟处理器,使用户可以在虚拟机中感受到多线程应用程序的性能优势。

④ 对虚拟 VLAN 的支持。为了更好地满足企业环境中的网络环境的定制,保证虚

拟机间信息的相互隔离,确保信息安全,在 Hyper-V 中,管理员可以通过虚拟机设置对虚拟化环境中的虚拟机划分 VLAN,以保证虚拟机间信息的相互隔离,确保信息安全。

⑤ 网络负载均衡。Hyper-V 中包含了全新的虚拟交换功能,这意味着虚拟机可用简单的方法配置运行 Windows 网络负载均衡(NLB)服务。Hyper-V 可在 NLB 群集中跨多个服务器为网络客户端服务器应用分配负载。NLB 对确保无状态应用(如在 Internet 信息服务(IIS)上运行的基于 Web 的应用)在工作负载增加时通过添加额外的服务器对其扩展极其有用。在负载增加时,NLB 允许添加额外的服务器来实现可扩展性。此外,NLB 还允许用户轻松替换故障服务器来实现可靠性。

⑥ 丰富的性能监控指标。为了更好地监控虚拟化平台中的宿主服务器和其上运行的虚拟机实例的性能状态,通过 Hyper-V 与 SCOM 相结合,管理员不仅对宿主服务器可以进行全方位的性能监控,而且可以同样高效细致地监控虚拟机的各方面性能。在减少管理员工作量的同时,高效地监控系统的运行状况。

⑦ 完整、开放的虚拟化扩展架构。为了企业未来的发展,Hyper-V 提供了良好的扩展开发框架和 API,以便企业能够将自行特有的硬件设备融入虚拟化平台中,为虚拟机提供虚拟化服务。Hyper-V 中包含的基于标准的 Windows 管理架构(WMI)接口以及 API 接口使软件供应商和开发人员可以快速创建自定义的工具、程序,并对虚拟化的平台进行改善。

(3) 微软针对 Hyper-V 的管理工具

微软通过 Microsoft System Center 来对虚拟机进行管理,包括以下三大组件。

① Operations Manager 2007,具备以下功能。

- 针对 IT 环境提供全面的监控。
- 涉及诸多操作系统和应用程序数以千计的事件跟踪和性能监视。
- 端到端的服务管理。

② Configuration Manager 2007,具备以下功能。

- 使得操作系统和应用程序部署更加安全、可靠。
- 配置管理,使得系统更加安全。
- 针对服务器、桌面、移动设备的全面资产管理。

③ Data Protection Manager 2007,具备以下功能。

- 提供一致的数据保护。
- 针对分公司的中央备份提供连续的数据保护。
- 针对数据中心备份的改进。
- 报告及监控功能。

(4) 通过 SCVMM 实现的功能

通过 SCVMM(System Center Virtual Machine Manager) 实现以下功能。

① 物理机到虚拟机的转换(P2V)。

② 虚拟机到虚拟机的转换(V2V)(分为 Virtual Server 到 Hyper-V,以及 VMware 到 Hyper-V)。

③ PowerShell 脚本(自动化数据中心)。

④ 可扩展控制台。

⑤ 支持虚拟机资源库。

⑥ Hyper-V 服务器场管理。

⑦ 模板/克隆。

此外,Hyper-V 的先天优势是可以兼容大量的驱动程序,而不必为虚拟机开发专用的驱动程序,只要设备能在 Windows Server 2008 下工作,那么 Hyper-V 虚拟机就能使用这些设备资源,再加上 Windows 驱动程序天生就比其他操作系统(如 Linux)的驱动程序丰富,因此在硬件支持上 Hyper-V 具有无可比拟的优势。所以 Hyper-V 最合适的应用场合就是单纯的微软服务器环境以及微软相关的服务,如 Active Directory、Exchange、SQL Server、SharePoint 等。这些微软本身的产品在 Hyper-V 下不但性能比其他的虚拟机产品更好,从兼容性和微软群集服务的设定以及管理维护上来说,也更具有优势。

但 Hyper-V 在 CPU 方面的限制是要求处理器必须支持 AMD-V 或者 Intel VT 技术,也就是说,处理器必须具备硬件辅助虚拟化技术。微软之前的 Virtual Server 2005 R2 不需要这个技术。对于 VMware 的产品来说,这也只是一个可选的特性,不像 Hyper-V 那样,是一个硬性的要求。处理器不支持 VT/AMD－V,就无法运行 Hyper-V。

在磁盘方面也有一定的限制:Hyper-V 虽然增加了 SCSI 控制器的支持,但是 Windows Server 2003 的虚拟机无法在 SCSI 磁盘上进行引导和安装。也就是说初次部署 Windows Server 2003 系统时,在 Hyper-V 虚拟机中不能使用 SCSI 虚拟机磁盘,只能用 IDE 虚拟机磁盘安装。

6. 服务器虚拟化的技术比较表

服务器虚拟化的技术比较见表 7-1。

表 7-1　服务器虚拟化的技术比较

厂商	技　　术	重点技术描述
IBM	支持硬分区 LPAR,软分区 DLPAR、动态微分区	基于动态微分区的 IBM Power 5 服务器已经可以在单处理器下做 10 次分区,同时可以执行 Linux 及 IBM 的 UNIX 操作系统。因此,虽然 AIX 的动态逻辑分区技术允许一台有 8 个处理器的服务器同时运行 8 个操作系统,并且能够在处理器之间转移工作负载,但虚拟引擎却可以创建微型分区,使一块 CPU 变成 10 台虚拟服务器
HP	支持基于 PA-RISC 平台的硬分区 nPAR,软分区 vPAR、vPAR 做在 nPar 上,只用在 PA-RISC 平台,不支持 Itanium 产品。声称即将支持基于 Itanium 的软分区 vmPAR	针对 Itanium 产品,计划在今年上半年推出基于 vmPAR 分割技术的产品,利用软件在单个 Itanium 处理器上切成最多 10 个分区(Partition),让 Integrity 的单一处理器可同时执行多操作系统,如 Windows、Linux、HP-UX 和 OpenVMS
SUN	支持硬分区 SUN domain、软分区 N1 Grid Container 或 Zone	Solaris 10 新增基于 N1 Grid Containers(N1 网格容器)的逻辑分区(Partition)功能,此功能允许在单一伺服主机上执行多种应用程序。这项技术将使用在 SUN 公司的 UltraSparc 处理器及 x86 处理器上

续表

厂商	技　术	重点技术描述
Microsoft	推行虚拟服务器企业级虚拟解决方案，可以实现多操作系统环境中服务器的整合和高效管理	推行虚拟服务器企业级虚拟解决方案，可以实现多操作系统环境中服务器的整合和高效管理。Virtual Server 是一种基于 Microsoft Windows 的服务器应用，使用户能够在同一种物理服务器上同时运行各种类型的操作系统，其中包括 Windows Server 2003、Windows 2000 Server、Linux、UNIX 和 OS/2。Virtual Server 是从 Connectix 的 Virtual PC 客户端虚拟软件产品线发展而来的，可以在符合业界标准的 Intel 服务器（IA32）上运行。Virtual Server 内建了虚拟计算机技术，使用软件和选定的硬件设备来创建一个模拟的运行环境
VMware	VMware 在操作系统和实现的硬件之间插入一个虚拟硬件。任何运行在基于 x86 上的系统都可以运行 VMware，其中包括所有版本的 Windows、Linux 和 NetWare	用户可以通过 VMware 的虚拟基础设施管理软件——vCenter 来管理由 ESX 服务器组成的池。该软件让管理员能够对资源进行切换，从而将未使用的容量减到最小，并且对服务器进行快速部署和维护。作为其 vCenter 的一部分，VMotion 可以将正在运行的虚拟机移植到一个不同的物理服务器上，并且不会对服务造成任何干扰，即使在没有硬件停机计划的情况下，也可以添加内存或进行硬件检修，还可以使用 VMotion 对负载进行随时随地的平衡操作

相比之下，对 UNIX 平台而言，IBM、HP、SUN 各有自己的技术标准，没有统一的虚拟化技术，因此，目前 UNIX 的虚拟化还受具体产品平台的制约。而 PC 服务器的虚拟化标准相对开放，无论是 VMware 还是 Microsoft Hyper-V，对 Windows 虚拟化都提供支持。

(1) 从技术来看，Hyper-V 的功能比 VMware 略为落后，表现在以下几方面

① 微软 Live Migration：是 VMware VMotion 的部分，VMware 从 2003 年将 VMotion 推向市场(微软落后了 7 年)。

② 集群文件系统：是 VMware VMFS 的部分，VMware 从 2001 年推出 VMFS(微软落后了 9 年)。

③ 32 个逻辑处理器：VMware 从 2006 年开始支持(微软落后了 4 年)。

④ 虚拟磁盘热添加：VMware 在 2006 年从 ESX 3 上开始支持(微软落后了 4 年)。

⑤ 虚拟磁盘热卸载：VMware 今天不支持(下一代的产品将提供)。

⑥ SLAT(内存虚拟化)：VMware 今天已经提供对 AMD RVI 和 Intel EPT 的支持(微软落后了 2 年)。

⑦ 动态内存：VMware 从 2001 年就开始支持(微软落后了 9 年)。

(2) 从功能上来看，VMware 与 Hyper-V 各有擅长

VMware 与 Hyper-V 功能对比见表 7-2。

7.2.2　网络虚拟化

网络虚拟化是目前业界关于虚拟化细分领域界定最不明确且存在争议较多的一个概念。但总体来说，分为纵向分隔和横向分隔两大类概念。

表 7-2　VMware 与 Hyper-V 功能对比表

比较项目	VMware	Hyper-V
高级内存管理：可以通过重新利用空闲内存和消除多个虚拟机上相同页面的重复内存来过量调配虚拟内存(相对物理内存而言)，从而最大限度提高内存利用率	是	无内存过量使用技术
热添加虚拟 CPU 和内存：可以在正在运行的虚拟机上热添加虚拟 CPU 和 RAM，从而在不挂起/关闭虚拟机的情况下，为应用程序增加资源	是	无法热添加虚拟 CPU/内存
实时存储迁移：无须关闭虚拟机即可将虚拟机文件从一个存储阵列实时迁移到另一个存储阵列。支持从一个存储协议到其他协议的此类迁移(例如，FC、iSCSI、NFS 和 DAS)	是	仅具有快速存储迁移功能，会导致虚拟机停机
虚拟机级别的容错功能：在出现硬件故障时，可以为虚拟机上的任何应用程序(而不仅仅是支持群集的应用程序)提供无停机保护	是	与第三方共同宣布尚未确定具体上市时间
逻辑资源池：可以从物理群集内分配逻辑资源池。还可以在不影响其他业务组的情况下，将这些资源分配给特定的业务组	是	不支持
群集级别的负载平衡：可以通过自动负载平衡和灵活的内存使用，安全地实现高水平的利用率。每个群集包含多达 32 个节点	是	缺少内存过量使用技术，严重限制了进行负载平衡时的灵活性
群集级别的电源管理：可在非峰值时段内将工作负载整合到更少的服务器上并关闭其他服务器，从而在群集级别节省电能。当常规工作时段的工作负载恢复时，将重新打开已关闭的服务器	是	非集成功能需要自定义脚本
群集级别的网络管理：通过创建能够跨整个群集并支持虚拟机移动性的分布式交换机，可以在群集级别管理虚拟交换机	是	不支持
群集级别的安全管理：可以在虚拟机级别(支持虚拟机移动性)管理和强制实施安全区域策略，而不必将特定物理主机分配到指定安全区域	是	不支持
专门构建的带小型代码库的精简体系结构：采用专门设计的体系结构，不但可以满足虚拟化要求(性能、扩展性)，而且还带有一个非常小的代码库，可最大限度缩小攻击范围并减少需要维护的代码量(安全性)	是	体系结构依赖于在父级分区运行的 Windows 2008 的大型子集
虚拟机虚拟资源纵向扩展：对于所有支持的客户服务器操作系统，每个虚拟机可以支持高达 8 个虚拟 CPU 和容量为 256GB 的 RAM，从而为高性能计算应用程序提供充足的计算资源	是	仅在 Windows 2008 虚拟机上支持 4 路 vSMP
广泛的客户操作系统支持：支持广泛的 Windows 客户服务器和客户端操作系统，甚至包括 Windows NT 4 和 DOS。支持所有主要的 Linux 发行版，包括半虚拟化的 Linux	是	支持大约 10 种操作系统
集成的内置网卡绑定功能：可以对主机上任何受支持的物理网卡集进行“绑定”，以在网卡出现故障时提供冗余。这种功能应内置于虚拟化平台中，而非需要使用第三方驱动程序，这些驱动程序只能与该特定第三方的物理网卡配合使用	是	依赖于只能在第三方网卡上工作的特定第三方驱动程序
带全面监控和警报功能的精简部署磁盘：可以精简部署磁盘，而无须预先分配所有存储空间。全面监控和警报功能，可防止意外用完所有物理存储空间	是	具有精简磁盘，但缺少全面的监控和警报工具

续表

比 较 项 目	VMware	Hyper-V
快照：可以对实时运行中的虚拟机拍摄快照，并在不关闭虚拟机的情况下将其恢复到快照点	是	无法在虚拟机开机时提交/回滚快照
在单个主机上同时执行实时迁移：可以在单个主机上执行并发实时迁移，以减少在计划内维护期间用于撤出单个主机上的所有虚拟机的时间	是	每台主机每次只能执行一项实时迁移
提供程序允许使用外部脚本编写工具与管理工具(如 Windows PowerShell、System Center Virtual Machine Manager 及其他第三方工具)管理 Hyper-V 服务器	不支持	支持
与 Microsoft Volume Shadow Copy 服务相集成	不支持	支持
Bitlocker 组件对硬盘文件实现加密保护	不支持	支持

1. 纵向分隔

早期的“网络虚拟化”，是指虚拟专用网络(VPN)。VPN 对网络连接的概念进行了抽象，允许远程用户访问组织的内部网络，就像物理上连接到该网络一样。网络虚拟化有助于保护 IT 环境，防止来自 Internet 的威胁，同时使用户能够快速安全地访问应用程序和数据。

随后的网络虚拟化技术随着数据中心业务要求发展为：多种应用承载在一张物理网络上，通过网络虚拟化分隔(称为纵向分隔，图 7-16)功能使得不同企业机构相互隔离，但可在同一网络上访问自身应用，从而实现了将物理网络进行逻辑纵向分隔虚拟化为多个网络。

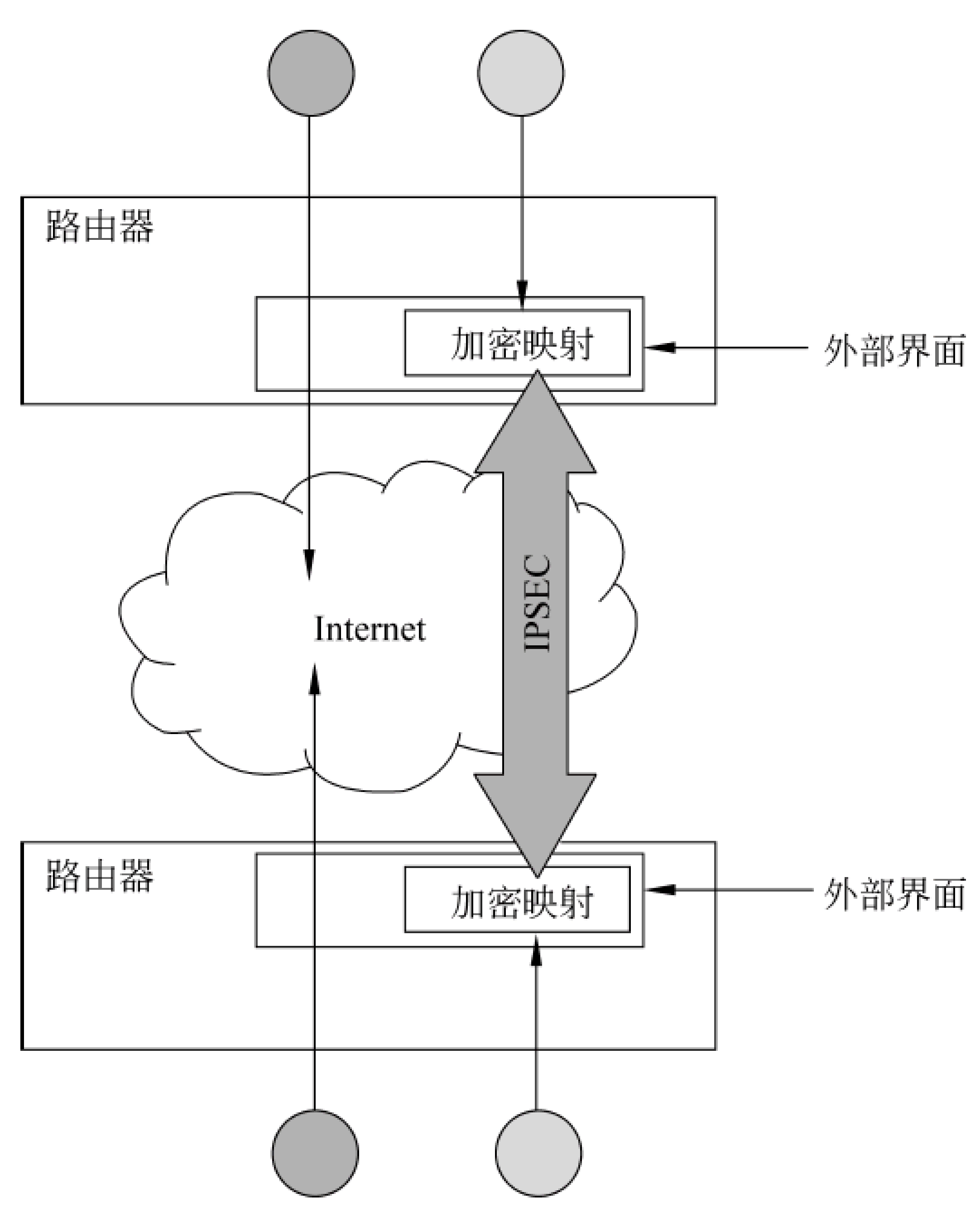


图 7-16　纵向分割示意图

如果把一个企业网络分隔成多个不同的子网络——它们使用不同的规则和控制，用户就可以充分利用基础网络的虚拟化路由功能，而不是部署多套网络来实现这种隔离机制。

网络虚拟化概念并不是什么新概念，因为多年来，虚拟局域网（VLAN）技术作为基本隔离技术已经广泛应用。当前在交换网络上通过 VLAN 来区分不同业务网段、配合防火墙等安全产品划分安全区域，是数据中心基本设计内容之一。

出于将多个逻辑网络隔离、整合的需要，VLAN、MPLS-VPN、Multi-VRF 技术在路由环境下实现了网络访问的隔离，虚拟化分隔的逻辑网络内部有独立的数据通道，终端用户和上层应用均不会感知其他逻辑网络的存在。但在每个逻辑网络内部，仍然存在安全控制需求，对数据中心而言，访问数据流从外部进入数据中心，则表明了数据在不同安全等级的区域之间流转，因此，有必要在网络上提供逻辑网络内的安全策略，而不同逻辑网络的安全策略有各自独立的要求，虚拟化安全技术，将一台安全设备可分隔成若干台逻辑安全设备（成为多个实例），从而很好地满足了虚拟化的深度强化安全要求。

2. 横向分隔

从另外一个角度来看，多个网络节点承载上层应用，基于冗余的网络设计带来复杂性，而将多个网络节点进行整合，虚拟化成一台逻辑设备，提升数据中心网络可用性、节点性能的同时将极大简化网络架构。

数据中心是企业 IT 架构的核心领域，传统的数据中心网络架构由于多层结构、安全区域、安全等级、策略部署、路由控制、VLAN 划分、二层环路、冗余设计等诸多因素，导致网络结构比较复杂，使得数据中心基础网络的运维管理难度较高。

使用网络虚拟化技术，用户可以将多台设备连接，“横向整合”起来组成一个“联合设备”，并将这些设备看作单一设备进行管理和使用。多个盒式设备整合类似于一台机架式设备，多台框式设备的整合相当于增加了槽位，虚拟化整合后的设备组成了一个逻辑单元，在网络中表现为一个网元节点，管理简单化、配置简单化、可跨设备链路聚合，极大简化了网络架构，同时进一步增强了冗余可靠性。

目前纵向和横向分隔的业界网络虚拟化倡导者为 Cisco 与 3Com 公司，两家业界巨头更希望能将任何基于服务的传统客户端/服务器安置到网络上，甚至在路由器中插入一张工作卡。该卡上带有一套全功能的 Linux 服务器，可以和路由器中枢相连。在这个 Linux 服务器中，可以安装诸如 Sniffer、VoIP、安全应用等，让路由器和交换机执行更多的服务。

但总的来说，目前网络虚拟化并无业界标准，成熟程度不如服务器虚拟化和存储虚拟化，而且相当一部分技术已经在设备采购中包含（比如 VPN、VLAN），无须单独统一部署。

7.2.3　存储虚拟化

随着人们对存储的需求越来越大。这样的需求刺激了各种新技术的出现，比如磁盘性能越来越好、容量越来越大。但是在大量的大中型信息处理系统中，单个磁盘无法满足需要，这样的情况下存储虚拟化技术就发展起来了。所谓虚拟存储，就是把多个存储介质模块（如硬盘、RAID）通过一定的手段集中管理起来，所有的存储模块在一个存储池（Storage Pool）中得到统一管理，从主机和工作站的角度看到的就不是多个硬盘，而是一个分区或者卷，就好像是一个超大容量（如 1T 以上）的硬盘。这种可以将多种、多个存储设备统一管理起来，为使用者提供大容量、高数据传输性能的存储系统，称为虚拟存储，如图 7-17 所示。

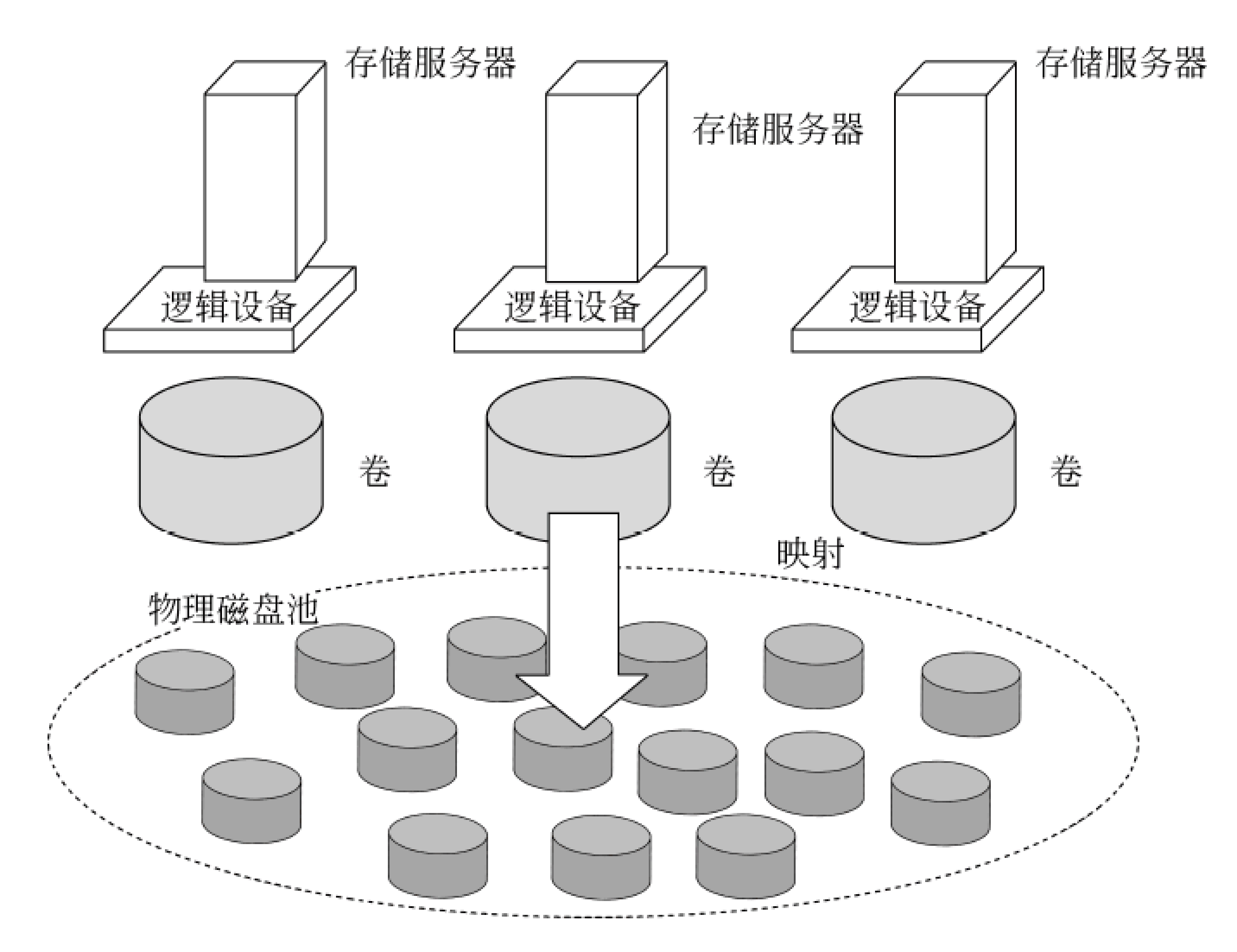


图 7-17　存储虚拟化架构图

虚拟存储设备主要通过大规模的 RAID 子系统和多个 I/O 通道连接到服务器上，智能控制器提供 LUN 访问控制、缓存和其他如数据复制等的管理功能。这种方式的优点在于存储设备管理员对设备有完全的控制权，而且通过与服务器系统分开，可以将存储的管理与多种服务器操作系统隔离，并且可以很容易地调整硬件参数。

目前虚拟存储的发展尚无统一标准，从虚拟化存储的拓扑结构来讲，主要有两种方式，即对称式（带内管理）与非对称式（带外管理）。对称式虚拟存储技术是指虚拟存储控制设备与存储软件系统、交换设备集成为一个整体，内嵌在网络数据传输路径中；非对称式虚拟存储技术是指虚拟存储控制设备独立于数据传输路径之外。

虚拟存储具有的特点如下。

（1）虚拟存储提供了一个大容量存储系统集中管理的手段，由网络中的一个环节（如服务器）进行统一管理，避免了由于存储设备扩充所带来的管理方面的麻烦。例如，使用一般存储系统，当增加新的存储设备时，整个系统（包括网络中的诸多用户设备）都需要重新进行烦琐的配置工作，才可以使这个“新成员”加入存储系统之中。而使用虚拟存储技术，增加新的存储设备时，只需要网络管理员对存储系统进行较为简单的系统配置更改，客户端无须任何操作，感觉上只是存储系统的容量增大了。

（2）虚拟存储可以大大提高存储系统的整体访问带宽。存储系统是由多个存储模块组成的，而虚拟存储系统可以很好地进行负载平衡，把每一次数据访问所需的带宽合理地分配到各个存储模块上，这样系统的整体访问带宽就增大了。例如，一个存储系统中有 4 个存储模块，每一个存储模块的访问带宽为 50Mbps，则这个存储系统的总访问带宽就可以接近各存储模块带宽之和，即 200Mbps。

（3）虚拟存储技术为存储资源管理提供了更好的灵活性，可以将不同类型的存储设备集中管理使用，保障了用户以往购买的存储设备的投资。

（4）虚拟存储技术可以通过管理软件，为网络系统提供一些其他有用的功能，如无须服务器的远程镜像、数据快照（Snapshot）等。

对于常规企业，存储虚拟化一般在性能相仿、存储零散分布的情况下适用。

业界虚拟存储产品主要有EMC公司的Invista、HDS公司的UPS和IBM公司的SVC。

(1) EMC公司：EMC公司2005年8月推出一款新型的存储路由器虚拟化产品。这款产品代号为Invista(见图7-18)的带外(Out of Band)专用设备是由两个组件构成的：一个双节点的服务器集群和一个与之相连接的光纤交换机。服务器上运行的软件均由EMC自行开发，主要负责审核每一个自主机送出的、经由光纤交换机传送到磁盘阵列的信息包。为每一个捕获到的信息包分配一个独一无二的标识符，对其进行分类，便于跨异构存储平台进行管理。

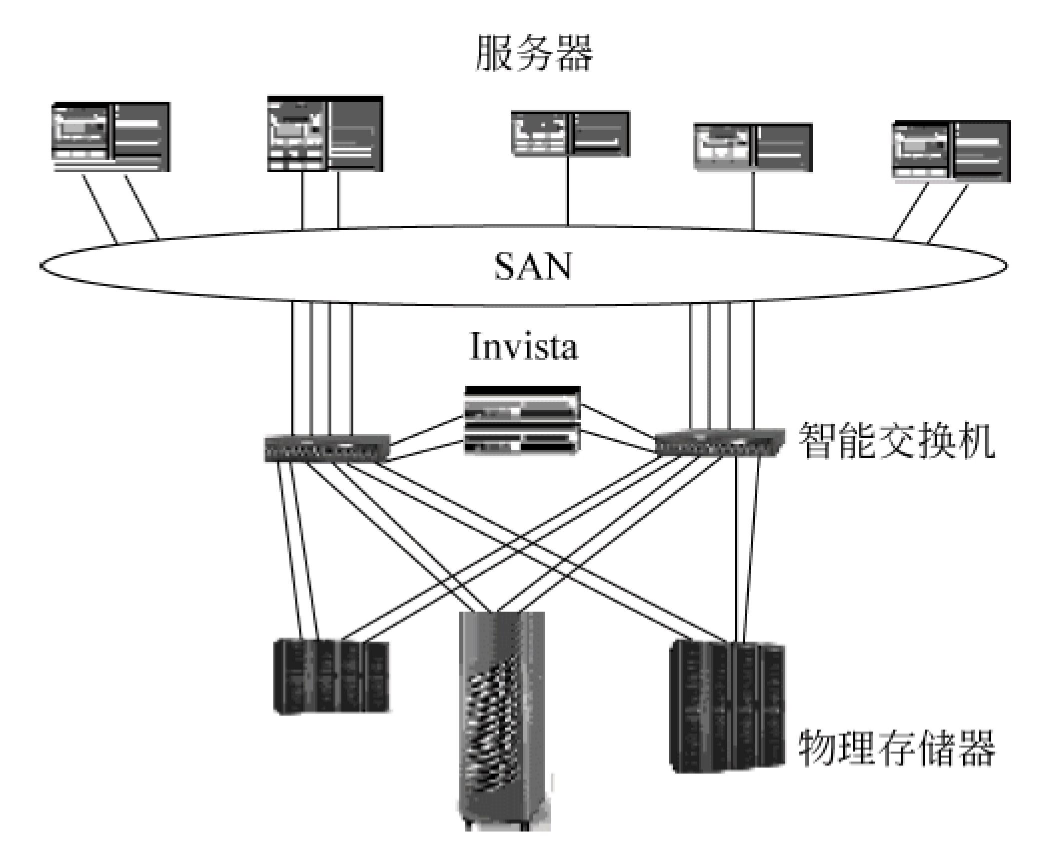


图7-18 Invista架构图

(2) HDS公司：2004年，HDS公司推出通用存储平台——TagmaStore Universal Storage Platform，即面向高端的虚拟化存储解决方案。HDS TagmaStore通用存储平台采用了虚拟化技术，通过内置的虚拟层，可以管理高达32PB的内部与外部存储容量，并支持在内部与外部存储中的逻辑分区，以及复杂的任意存储地点间的远程复制功能。

2005年，HDS坚持走基于磁盘控制器的虚拟化道路，2005年7月推出的Hitachi TagmaStore网络存储控制器NSC55就是典型的代表。NSC55采用了Hitachi Universal Star Network交换架构，提供由TagmaStore通用存储平台带来的、经过验证的企业级功能——外部存储虚拟化、逻辑资源分区和通用复制。NSC55集高端存储平台才具有的高级功能以及已经过用户验证的虚拟化软件两大功能于一身，打破了目前模块化产品的局限性，超出所有现有模块化“磁盘式存储”以及基于交换机或其他设备的虚拟化功能，是存储行业的一大创举。

(3) IBM公司：IBM公司2003年推出的基于网络的存储虚拟化产品主要包括SAN Volume Controller(SVC)和SAN File System。SVC是整个SAN(存储区域网)网络的控制器，可以将SAN中的各种存储设备整合成一个存储池，并按需分配存储空间、性能和功能。SVC对服务器和存储设备都是透明的。SVC为各种不同的存储设备提供了一个统一的数据复制平台。SVC是一个软硬件集成的产品。SVC刚推出时，可实现存储虚拟化的软件运行在一个类似服务器的硬件设备上。如今，此软件也可以安装在SAN中的交换机上，

如思科的 MDS 网络交换机。SVC 是为一个完全开放的存储环境设计的，可以兼容各种不同的存储设备。

7.2.4 应用虚拟化

应用虚拟化通常包括两层含义，一是应用软件的虚拟化，二是桌面的虚拟化。所谓的应用软件虚拟化，就是将应用软件从操作系统中分离出来，通过自己压缩后的可执行文件夹来运行，而无须任何设备驱动程序或者与用户的文件系统相连。借助于这种技术，用户可以减小应用软件的安全隐患和维护成本，并进行合理的数据备份与恢复。除了可以将应用软件与操作系统分离外，一部分解决方案还可以将应用软件流水化包装起来，应用软件无须安装，只要一部分程序能够在计算机上运行即可。用户只需使用他们自己需要的那部分程序或者功能即可。

应用软件虚拟化技术至今仍然不是很成熟，相对而言，桌面虚拟化则要成熟得多。桌面虚拟化就是专注于桌面应用及其运行环境的模拟与分发，是对现有桌面管理自动化体系的完善和补充。

当今的桌面环境将桌面组件（硬件、操作系统、应用程序、用户配置文件和数据）联系在一起，给支持和维护工作带来了很大困难。因此，一个层发生问题往往会导致连锁反应，最终破坏整个桌面，使用 IT 部门恢复本地存储的用户数据和设置非常困难并且成本昂贵。

个人计算设备和操作系统的繁多使用户/服务器端应用的测试、调试和客户定制开发成本高昂且耗费时间。

采用桌面虚拟化技术（图 7-19）之后，将不需要在每个用户的桌面上部署和管理多个软件客户端系统，所有应用客户端系统都将一次性地部署在数据中心的一台专用服务器上，这台服务器就放在应用服务器的前面。客户也将不需要通过网络向每个用户发送实际的数据，只有虚拟的客户端界面（屏幕图像更新、按键、鼠标移动等）被实际传送并显示在用户的计算机上。这个过程对最终用户是一目了然的，最终用户的感觉好像是实际的客户端软件正在他的桌面上运行一样。

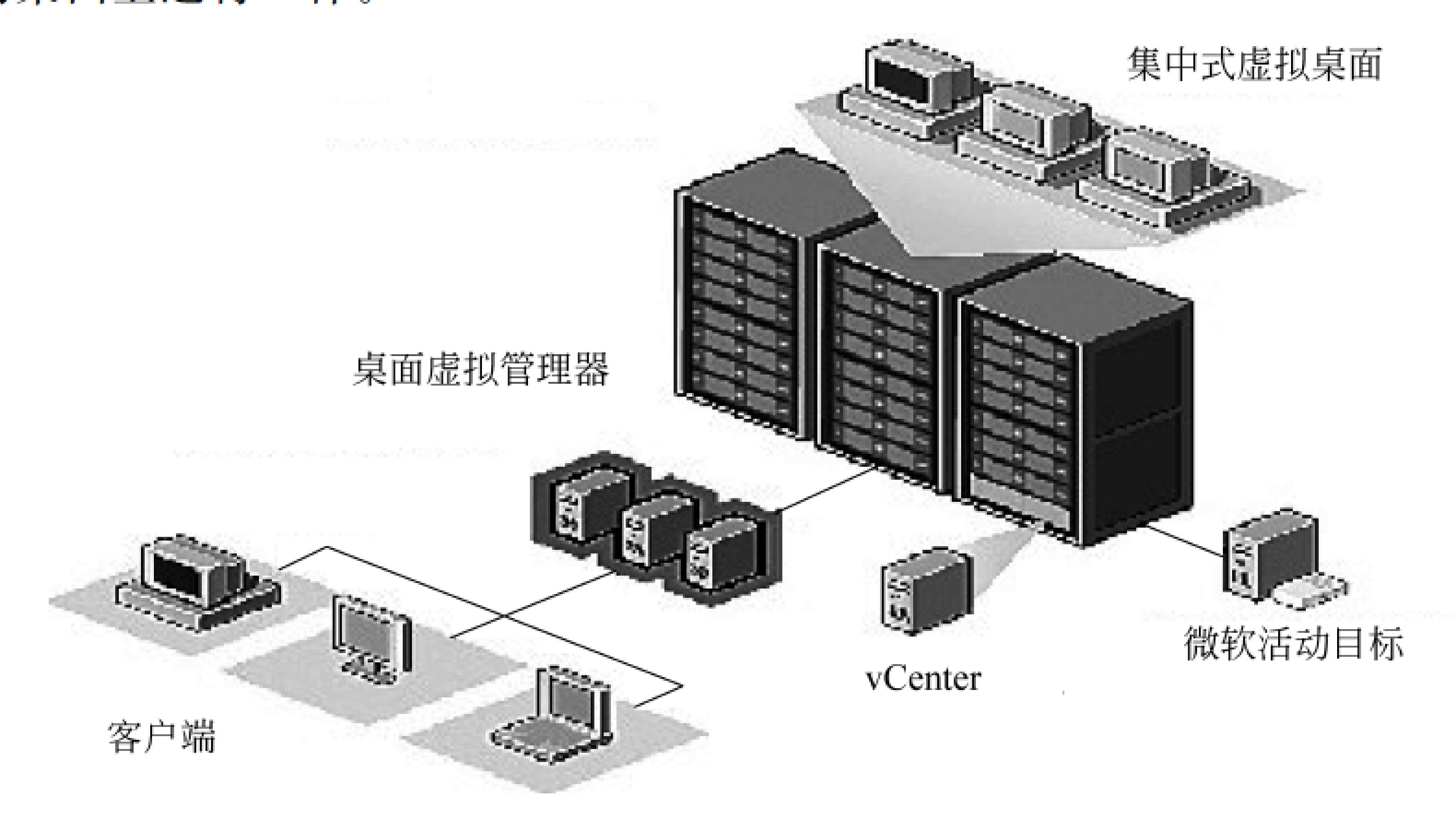


图 7-19 桌面虚拟化架构图

传统的用户/服务器端应用要求在每个用户的计算机上安装客户端软件，从而导致更高的成本，因为需要在分布式网络上管理这些软件的部署、补丁和升级。这个问题随着用户登录到每个新应用系统的需求量呈增长趋势，因为 IT 部门需要在每个用户的桌面上部署另一个独特的客户端设备。即便在最讲究战术的接入服务场景中，应用虚拟化可以带来的成本效益也是相当诱人的。通过将 IT 系统的管理集中起来，企业能够同时实现各种不同的效益，从带宽成本节约到提高 IT 效率和员工生产力以及延长陈旧的或当前系统的寿命等，最主要有以下三点。

1. 数据安全

由于应用虚拟化软件只是将运行的图像更新通过网络传输和显示在远程设备的显示设备上，数据与文件不会通过网络进行传输。而且由于只需要开一个端口，网络安全性也大大提高。

即使不怀好意者截取数据包，由于图像的差异化，也很难恢复成原来的数据，利用加密手段将使整个系统的安全性得到大大提高。

2. 高效管理

应用虚拟化软件，将操作系统的安装、运行环境与用户实际的操作环境进行分离，从而实现了操作系统的管理和使用的分离。实现了便捷、完整的桌面生命周期管理。

(1) 管理员只需要分配新的账号，新用户就可以立即通过任何设备访问属于他的桌面系统。

(2) 管理员通过配置，可以使用户通过相同设备同时访问多个桌面系统。

(3) 管理员只需要对后台镜像统一打一次补丁，所有用户下次使用的桌面就是最新的状态。

(4) 管理员在后台镜像安装一次程序，下次用户登录桌面，就可以使用此应用。

3. 降低 TCO

(1) 减少投资成本：可以采用瘦用户端，投资成本只有传统 PC 的 50%，同时延长所有设备的使用寿命，能够将年折旧减少 50%。

(2) 减少运维成本：通常一个专业机房管理员能够管理 90 台桌面，而使用应用虚拟化软件，一个管理员可以管理几乎所有的桌面，运维成本大大下降。

(3) 降低运行成本：瘦客户端的功率接近传统 PC 的 1/6，使用虚拟化软件可以减少约 80%的电量消耗。

目前，主流的应用虚拟化厂商有 Citrix 公司的 XenDesktop、VMware 公司的 View(以前的 VMware VDI)、微软公司的 App-V(以前的 Soft Grid)。

目前看来，3 大厂商在这个层面采用了不同的拆分技术。VMware 公司采用物理的拆分方法，即基于服务器的差异磁盘的技术，实现差异的镜像，比如 200 个用户可以使用一个共同的“母盘”镜像，每个用户自己的差异信息，包括应用(VMware 自己的应用虚拟化 Thin App 实际是一个打包方法，需要存储在本地，如图 7-20 所示)与配置信息，使用时将两者结合起来提供服务，这种完全基于二进制的拆分方法是典型的服务器虚拟化厂商的技术，大大降低了存储量。但是这种技术仍然要求管理员一定程度上对每个用户的镜像进行管理，因为仍然存在这种一一对应。

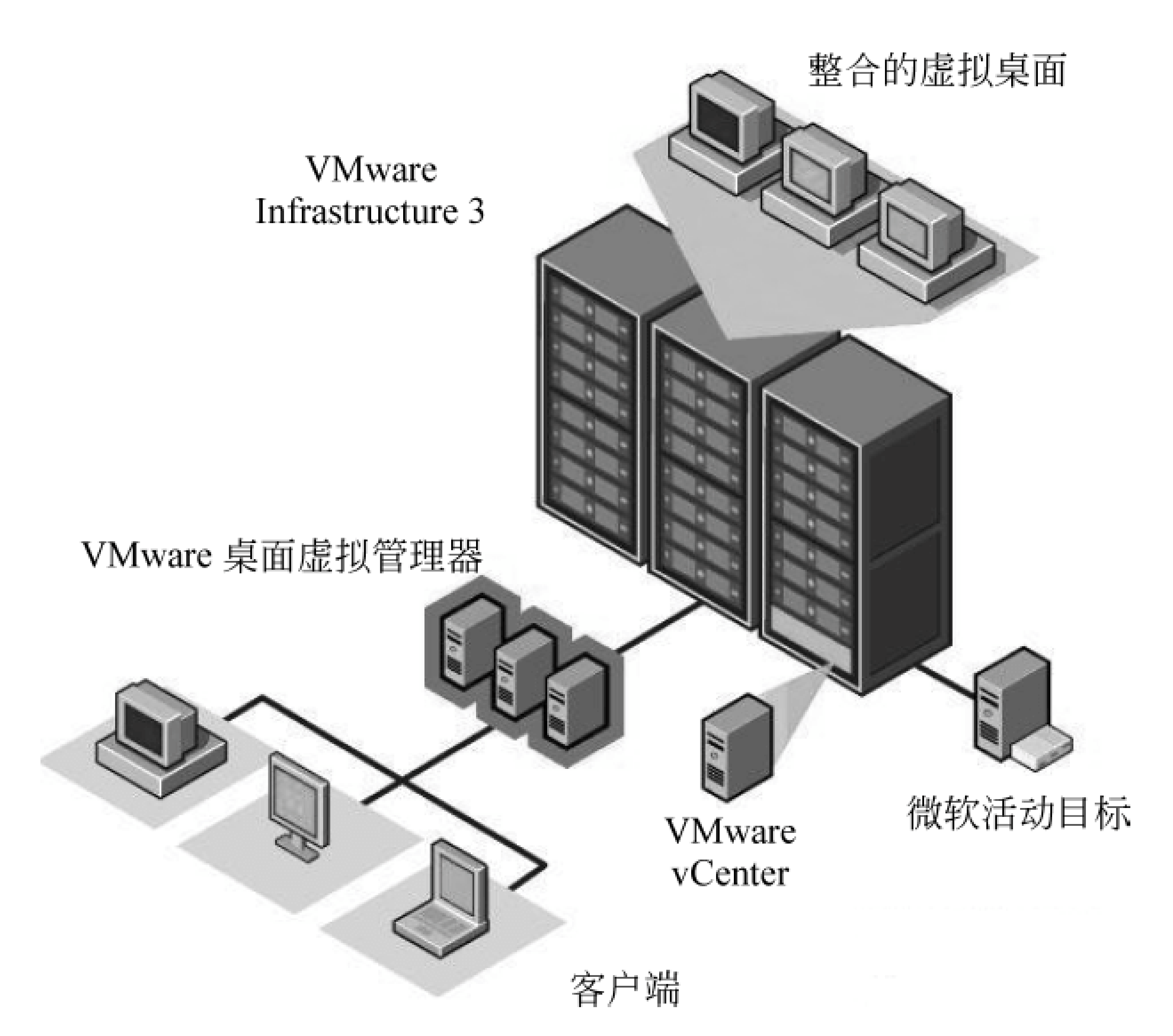


图 7-20　VMware 应用虚拟化技术

而 Citrix 作为应用虚拟化的传统厂商，则采用了自己很成熟的“逻辑”拆分法，按照逻辑分类将其拆分，即按操作系统、应用与配置文件进行拆分，使用时按需进行组装，这样能够保证不同逻辑单元的相互独立性，防止一方发生变化对其他方面造成的影响，例如应用与系统的升级和维护如图 7-21 所示。

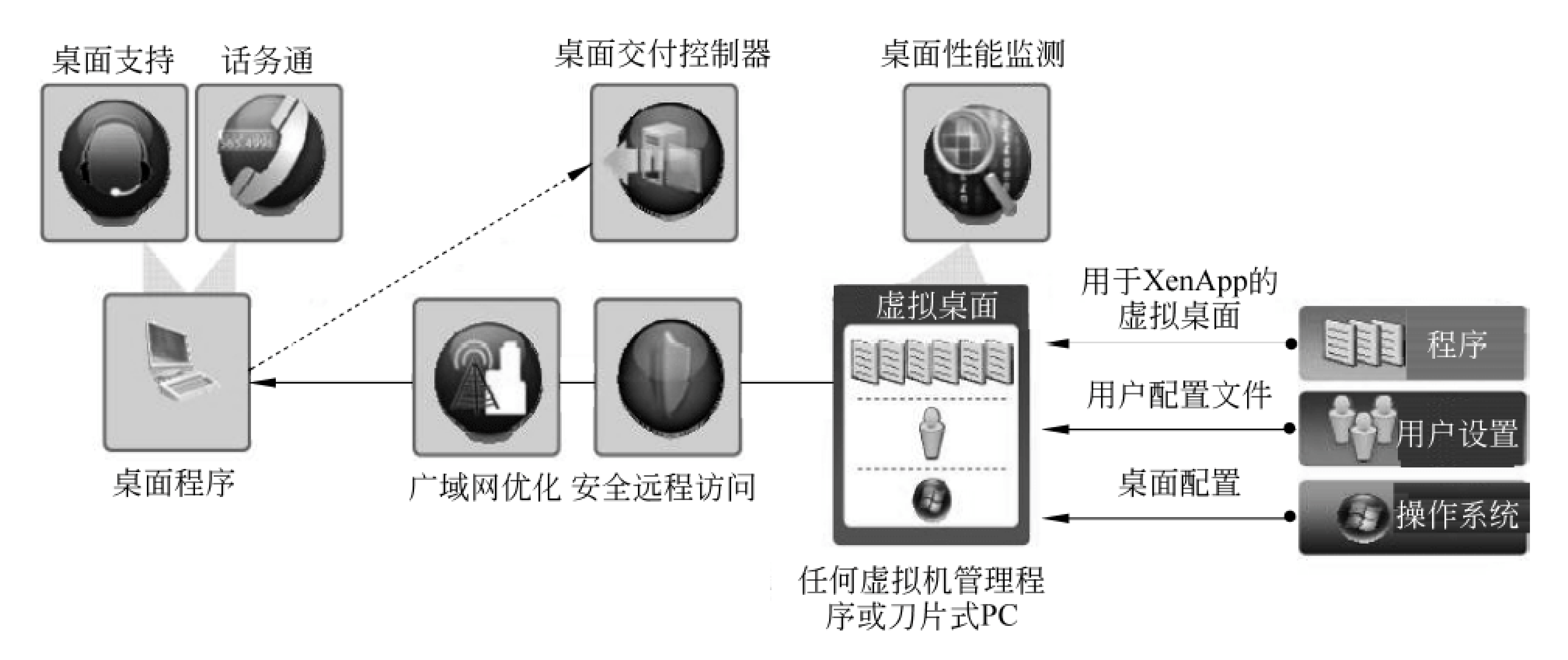


图 7-21　Citrix 应用虚拟化技术

微软则介于两者之间，根据官方的介绍，用户可以把自己制作好的虚拟机上传到服务器上，可以看到是一个用户与镜像一一对应的管理方法。当然微软自己具有终端服务和 RDP，可以采用与 Citrix 一样的方法；而它又有 Soft Grid(现在叫 App-V，类似于 Thin App 的打包技术)与虚拟服务器的差异磁盘技术，也可以采用 VMware 的技术路线。

7.2.5　虚拟化技术的比较

综上所述，对各种虚拟化技术比较如表 7-3 所示。

表 7-3　各种虚拟化技术的比较

比较项目	服务器虚拟化	存储虚拟化	网络虚拟化	应用虚拟化
产生年代	20 世纪 60 年代	2003 年	20 世纪末期	21 世纪
成熟程度	高	中	低	低
主流厂商	VMware Microsoft IBM HP	EMC HDS IBM	Cisco 3Com	Citrix VMware Microsoft
增强管理性	高	中	中	高
可靠性	高	中	中	中
可用性	高	高	中	高
兼容性	高	中	低	中
可扩展性	高	高	中	中
部署难度	中	高	中	高

在这四种虚拟化技术中，服务器虚拟化技术、应用虚拟化中的桌面虚拟化技术相对成熟，也是使用较多的技术而其他虚拟化技术则还需要在实践中进一步检验和完善。

7.3　虚拟化架构对比分析

云计算平台需要有资源池为其提供能力输出，这种能力包括计算能力、存储能力和网络能力，为了将这些能力调度到其所需要的地方，云计算平台还需要对能力进行调度管理，这些能力均是由虚拟化资源池提供的。

云计算离不开底层的虚拟化技术支持。维基百科列举的虚拟化技术超过 60 种，基于 X86(CISC)体系的超过 50 种，也有基于 RISC 体系的，其中有 4 种虚拟化技术是当前最为成熟而且应用最为广泛的，分别是 VMware 的 ESX、微软的 Hyper-V、开源的 XEN 和 KVM。云计算平台选用何种虚拟化技术将是云计算建设所要面临的问题。

形成资源池计算能力的物理设备可能有两种，一种是基于 RISC 的大/小型计算机；另一种是基于 CISC 的 x86 服务器。大/小型计算机通常意味着高性能、高可靠性和高价格，而 x86 服务器与之相比有些差距，但随着 Intel 和 AMD 等处理器厂商技术的不断发展，原本只在小型计算机上才有的技术已经出现在了 x86 处理器上，如 64 位技术、虚拟化技术、多核心技术等，使得 x86 服务器在性能上突飞猛进。通过 TPC 组织在 2011 年 3 月所公布的单机计算机性能排名中可以看出，4 路 32 核的 x86 服务器性能已经位列前 10 名，更重要的是 x86 服务器的性价比相对小型计算机有约 5 倍的优势。因此，选择 x86 服务器作为云计算资源池，更能凸显出云计算的低成本优势。

由于单机计算机的处理能力越来越大，以单机资源为调度单位的颗粒度就太大了，因此需要有一种技术让资源的调度颗粒更细小，使资源得到更有效和充分的利用，这就引入了虚

拟化技术。

从虚拟化的实现方式来看，虚拟化架构主要有两种形式：宿主架构和裸金属架构。在宿主架构中的虚拟机作为主机操作系统的一个进程来调度和管理，裸金属架构下则不存在主机操作系统，它是以管理程序（Hypervisor）直接运行在物理硬件之上，即使是有类似主机操作系统的父分区或 Domain 0，也是作为裸金属架构下的虚拟机存在的。宿主架构通常用于 PC 上的虚拟化，如 Windows Virtual PC、VMware Workstation、Virtual Box、Qemu 等，而裸金属架构通常用于服务器的虚拟化，如文中提及的 4 种虚拟化技术。

1. ESX 的虚拟化架构

ESX 是 VMware 的企业级虚拟化产品，2001 年开始发布 ESX 1.0，到 2011 年 2 月发布 ESX 4.1 Update 1。

ESX 服务器启动时，首先启动 Linux Kernel，通过这个操作系统加载虚拟化组件，最重要的是 ESX 的 Hypervisor 组件，称为 VMkernel，VMkernel 会从 LinuxKernel 完全接管对硬件的控制权，而该 Linux Kernel 作为 VMkernel 的首个虚拟机，用于承载 ESX 的服务器控制台（Service Console），实现本地的一些管理功能。

VMkernel 负责为所承载的虚拟机调度所有的硬件资源，但不同类型的硬件会有区别。

虚拟机对于 CPU 和内存资源通过 VMkernel 直接访问，最大限度地减少了开销，CPU 的直接访问得益于 CPU 硬件辅助虚拟化（Intel VT-x 和 AMD AMD-V，第一代虚拟化技术），内存的直接访问得益于 MMU（内存管理单元，属于 CPU 中的一项特征）硬件辅助虚拟化（Intel 公司的 EPT 和 AMD 公司的 RVI/NPT，第二代虚拟化技术）。

虚拟机对于 I/O 设备的访问则有多种方式，以网卡为例，有两种方式可供选择：一是利用 I/O MMU 硬件辅助虚拟化（Intel VT-d 和 AMD-Vi）的 VMDirectPath I/O，使得虚拟机可以直接访问硬件设备，从而减少对 CPU 的开销；二是利用半虚拟化的设备 VMXNETx，网卡的物理驱动在 VMkernel 中，在虚拟机中装载网卡的虚拟驱动，通过这两者的配对来访问网卡，与仿真式网卡（IntelE1000）相比有着较高的效率。半虚拟化设备的安装是由虚拟机中的 VMware tool 来实现的，可以在 Windows 虚拟机的右下角找到它。网卡的这两种方式，前者有着显著的先进性，但后者用得更为普遍，因为 VMDirectPath I/O 与 VMware 虚拟化的一些核心功能不兼容，如热迁移、快照、容错、内存过量使用等。

ESX 的物理驱动是内置在 Hypervisor 中，所有设备驱动均是由 VMware 预植入的。因此，ESX 对硬件有严格的兼容性列表，不在列表中的硬件，ESX 将拒绝在其上面安装。

2. Hyper-V 的虚拟化架构

Hyper-V 是微软新一代的服务器虚拟化技术，首个版本于 2008 年 7 月发布，目前最新版本是 2011 年 4 月发布的 R2 SP1 版。Hyper-V 有两种发布版本：一是独立版，如 Hyper-V Server 2008，以命令行界面实现操作控制，是一个免费的版本；二是内嵌版，如 Windows Server 2008，Hyper-V 作为一个可选开启的角色。

对于一台没有开启 Hyper-V 角色的 Windows Server 2008 来说，这个操作系统将直接操作硬件设备，一旦在其中开启了 Hyper-V 角色，系统会要求重新启动服务器。虽然重启后的系统在表面看来没什么区别，但从体系架构上看则与之前的完全不同了。在这次重启动过程中，Hyper-V 的 Hypervisor 接管了硬件设备的控制权，先前的 Windows Server 2008 则成为 Hyper-V 的首个虚拟机，称之为父分区，负责其他虚拟机（称为子分区）以及 I/O 设

备的管理。Hyper-V 要求 CPU 必须具备硬件辅助虚拟化,但对 MMU 硬件辅助虚拟化则是一个增强选项。

其实 Hypervisor 仅实现了 CPU 的调度和内存的分配,而父分区控制着 I/O 设备,它通过物理驱动直接访问网卡、存储设备等。子分区要访问 I/O 设备需要通过子分区操作系统内的 VSC(虚拟化服务客户端),对 VSC 的请求由 VMBus(虚拟机总线)传递到父分区操作系统内的 VSP(虚拟化服务提供 者),再由 VSP 重定向到父分区内的物理驱动。每种 I/O 设备均有各自的 VSC 和 VSP 配对,如存储设备、网络、视频和输入设备等,整个 I/O 设备访问过程对 于子分区的操作系统是透明的。其实在子分区操作系统内,VSC 和 VMBUS 就是作为 I/O 设备的虚拟驱动,它是子分区操作系统首次启动时由 Hyper-V 提供的集成服务安装包,这也算是一种半虚拟化的设备,使得虚拟机与物理 I/O 设备无关。如果子分区的操作系统没有安装 Hyper-V 集成服务包或者不支持 Hyper-V 集成服务包(对于这种操作系统,微软称为 Unenlightened OS,如未经认证支持的 Linux 版本和旧的 Windows 版本),则这个子分区只能运行在仿真状态。其实微软所宣称的启蒙式(Enlightenment)操作系统,就是支持半虚拟化驱动的操作系统。

Hyper-V 的 Hypervisor 是一个非常精简的软件层,不包含任何物理驱动,物理服务器的设备驱动均是驻留在父分区的 Windows Server 2008 中,驱动程序的安装和加载方式与传统 Windows 系统没有任何区别。因此,只要是 Windows 支持的硬件,就都能被 Hyper-V 所兼容。

3. XEN 的虚拟化架构

XEN 最初是剑桥大学 Xensource 的一个开源研究项目,2003 年 9 月发布了首个版本 XEN 1.0,2007 年 Xensource 被 Citrix 公司收购,开源 XEN 转由 www.xen.org 继续推进,该组织成员包括个人和公司(如 Citrix、Oracle 等)。目前该组织在 2011 年 3 月发布了最新版本 XEN 4.1。

相对于 ESX 和 Hyper-V 来说,XEN 支持更广泛的 CPU 架构,前两者只支持 CISC 的 x86/x86_64 CPU 架构,XEN 除此之外还支持 RISC CPU 架构,如 IA64、ARM 等。

XEN 的 Hypervisor 是服务器经过 BIOS 启动之后载入的首个程序,然后启动一个具有特定权限的虚拟机,称为 Domain 0(简称为 Dom 0)。Dom 0 的操作系统可以是 Linux 或 UNIX,Domain 0 实现对 Hypervisor 的控制和管理功能。在所承载的虚拟机中,Dom 0 是唯一可以直接访问物理硬件(如存储和网卡)的虚拟机,它通过本身加载的物理驱动,为其他虚拟机(Domain U,简称 DomU)提供访问存储设备和网卡的桥梁。

XEN 支持两种类型的虚拟机,一类是半虚拟化(Para-Virtualization,PV),另一类是全虚拟化(XEN 称其为 HVM,Hardware Virtual Machine)。半虚拟化需要特定内核的操作系统,如基于 Linux paravirt_ops(Linux 内核的一套编译选项)框架的 Linux 内核,而 Windows 操作系统由于其封闭性则不能被 XEN 的半虚拟化所支持,XEN 的半虚拟化有个特别之处就是不要求 CPU 具备硬件辅助虚拟化,这非常适用于 2007 年之前的旧服务器虚拟化改造。全虚拟化支持原生的操作系统,特别是针对 Windows 这类操作系统,XEN 的全虚拟化要求 CPU 具备硬件辅助虚拟化,它修改的 Qemu 仿真所有硬件,包括 BIOS、IDE 控制器、VGA 显示卡、USB 控制器和网卡等。为了提升 I/O 性能,全虚拟化特别针对磁盘和网卡采用半虚拟化设备来代替仿真设备,这些设备驱动称之为 PV on HVM。为了使 PV

on HVM 有最佳性能,CPU 应具备 MMU 硬件辅助虚拟化。

XEN 的 Hypervisor 层非常薄,少于 15 万行的代码量,不包含任何物理设备驱动程序,这一点与 Hyper-V 是非常类似的,物理设备的驱动程序均是驻留在 Dom 0 中,可以重用现有的 Linux 设备驱动程序。因此,XEN 对硬件兼容性也是非常广泛的,Linux 支持的,它就支持。

4. KVM 的虚拟化架构

KVM 的全称是 Kernel-based Virtual Machine,意思是基于内核虚拟机。其最初是由 Qumranet 公司开发的一个开源项目,2007 年 1 月首次被整合到 Linux 2.6.20 核心中;2008 年,Qumranet 被 RedHat 所收购,但 KVM 本身仍是一个开源项目,有 RedHat、IBM 等厂商支持。KVM 作为 Linux 内核中的一个模块,与 Linux 内核一起发布,至 2011 年 1 月的最新版本是 KVM-KMOD 2.6.37。

与 XEN 类似,KVM 支持广泛的 CPU 架构,除了 x86/x86_64 CPU 架构之外,还将会支持大型计算机(S/390)、小型计算机(PowerPC、IA64)及 ARM 等。

KVM 充分利用了 CPU 的硬件辅助虚拟化能力,并重用了 Linux 内核的诸多功能,使得 KVM 本身非常瘦小,KVM 的创始者 AviKivity 声称 KVM 模块仅有约 10000 行代码,但不能认为 KVM 的 Hypervisor 就是这个代码量,因为从严格意义来说,KVM 本身并不是 Hypervisor,它仅是 Linux 内核中的一个可装载模块,其功能是将 Linux 内核转换成一个裸金属的 Hypervisor。这相对于其他裸金属架构来说,它是非常特别的,有些类似于宿主架构,业界甚至有人称其是半裸金属架构。

通过 KVM 模块的加载将 Linux 内核转变成 Hypervisor,KVM 在 Linux 内核的用户(User)模式和内核(Kernel)模式基础上增加了客户(Guest)模式。Linux 本身运行于内核模式,主机进程运行于用户模式,虚拟机则运行于客户模式,使得转变后的 Linux 内核可以将主机进程和虚拟机进行统一的管理和调度,这也是 KVM 名称的由来。

KVM 利用修改的 QEMU 提供 BIOS、显卡、网络、磁盘控制器等的仿真,但对于 I/O 设备(主要指网卡和磁盘控制器)来说,则必然带来性能低下的问题。因此,KVM 也引入了半虚拟化的设备驱动程序,通过虚拟机操作系统中的虚拟驱动与主机 Linux 内核中的物理驱动程序相配合,提供近似原生设备的性能。由此可以看出,KVM 支持的物理设备也即是 Linux 所支持的物理设备。

本书所介绍的 4 种虚拟化技术都用到了半虚拟化驱动程序,若要在不同虚拟化架构之间迁移虚拟机,这些半虚拟化驱动程序必然会带来兼容性问题。因此,RedHat 和 IBM 联合 Linux 社区推出 VirtIO 半虚拟化驱动程序开发标准,基于 VirtIO 的半虚拟化驱动程序独立于 Hypervisor。跨平台迁移时半虚拟化驱动程序仍可重用,使得不同虚拟化架构之间更容易实现互操作。

目前,传统概念下的半虚拟化和全虚拟化的界限越来越模糊了,而且半虚拟化和全虚拟化得到了有机的整合,如半虚拟化的设备驱动程序和全虚拟化的虚拟机在上述 4 种虚拟化架构中得到了统一,很多虚拟化厂商也不再明确自己的虚拟化产品归类(如 VMware 和微软)。

随着 CPU 硬件辅助虚拟化技术发展到了二代,新版的操作系统对虚拟化技术的原生支持(如 Windows 7 的 Natively Enlightened,Linux 的 paravirt_ops 内核选项),使

Hypervisor 对虚拟机的 CPU 调度和内存管理的干预越来越少。软件做得越少而硬件做得越多,如虚拟机之间内存管理所需用到的地址翻译由软件的影式分页(Shadow Paging)转变为由 CPU 硬件加速的嵌套分页(Nested Paging),各种虚拟化技术既有全虚拟化技术对操作系统的兼容性,又有半虚拟化技术所带来的性能优势。

从架构上来看,各种虚拟化技术没有明显的性能差距,稳定性也在逐渐逼近中,各自有着自身的优势场景和市场群体。因此,在进行虚拟化技术选型时,不应局限于某一种虚拟化技术,而应该有一套综合管理平台实现对各种虚拟化技术的兼容并蓄,实现不同技术架构的统一管理及跨技术架构的资源调度,最终达到云计算可运营的目的。

7.4　虚拟平台所面临的安全问题

随着虚拟化技术不断向前发展,许多单位都会面对实施虚拟化的诱人理由,如服务器的整合、更快的硬件、使用上的简单、灵活的快照技术等,这都使得虚拟化更加引人注目。在有些机构中,虚拟化已经成为其架构中的重要组成部分。在这里,技术再次走在了最佳的安全方法的前面。随着机构对灾难恢复和业务连续性的重视,特别是在金融界,虚拟环境正变得越来越普遍。人们应该关注这种繁荣背后的隐忧。

7.4.1　使用虚拟化环境时存在的缺陷

(1) 如果主机受到破坏,那么主要的主机所管理的客户端服务器有可能被攻克。

(2) 如果虚拟网络受到破坏,那么客户端也会受到损害。

(3) 需要保障客户端共享和主机共享的安全,因为这些共享有可被不法之徒利用的漏洞。

(4) 如果主机有问题,那么所有的虚拟机都会产生问题。

(5) 虚拟机被认为是二级主机,它们具有类似的特性,并以与物理机的类似的方式运行。在以后的几年中,虚拟机和物理机之间的不同点将会逐渐减少。

(6) 在涉及虚拟领域时,最少特权技术并没有得到应有的重视,甚至遭到了遗忘。这项技术可以减少攻击面,并且应当在物理的和类似的虚拟化环境中采用这项技术。

7.4.2　保障虚拟服务器环境安全的措施

(1) 升级操作系统和应用程序,应当在所有的虚拟机和主机上进行。主机应用程序应当少之又少,仅安装所需要的程序即可。

(2) 在不同的虚拟机之间,用防火墙进行隔离和防护,并确保只能处理经许可的协议。

(3) 使每一台虚拟机与其他的虚拟机和主机相隔离,尽可能地在所有方面都进行隔离。

(4) 在所有的主机和虚拟机上安装和更新反病毒机制,因为虚拟机如同物理机器一样易受病毒和蠕虫的感染。

(5) 在主机和虚拟机之间使用 IPSEC 或强化加密,因为虚拟机之间、虚拟机与主机之间的通信可能被嗅探和破坏。虽然厂商们在想方设法改变这种状况,但目前这仍是一真实的威胁。企业仍需要最佳的方法来对机器之间的通信实施加密。

(6) 不要从主机浏览互联网,间谍软件和恶意软件所造成的感染仍有可能危害主机。记住,主机管理着虚拟机,发生在虚拟机上的问题会导致严重的问题和潜在的"宕机"时间、服务的丧失等。

(7) 在主机上保障管理员和管理员组账户的安全,因为未授权用户对特权账户的访问能导致严重的安全损害。调查发现,主机上的管理员(根)账户不如虚拟机上的账户安全。记住,安全性是由最弱的登录点决定的。

(8) 强化主机操作系统,并终止和禁用不必要的服务。保持操作系统的精简,可以减少被攻击的机会。

(9) 关闭不使用的虚拟机。如果不需要虚拟机,就不要运行它。

(10) 将虚拟机整合到企业的安全策略中。

(11) 保证主机的安全,确保在虚拟机离线时,非授权用户无法破坏虚拟机文件。

(12) 采用可隔离虚拟机管理程序的方案,这些系统可以进一步隔离和更好地保障虚拟环境的安全。

(13) 确保主机驱动程序的更新和升级,这会保障硬件以最优的速度运行,而且软件的更新可极大地减少漏洞利用和拒绝服务攻击的机会。

(14) 要禁用虚拟机中未用的端口。如果虚拟机环境并不利用端口技术,就应当禁用它。

(15) 监视主机和虚拟主机上的事件日志和安全事件。这些日志应当妥善保存,用于日后的安全审计。

(16) 限制并减少硬件资源的共享。从某种意义上讲,安全与硬件资源共享如同鱼与熊掌,不可兼得。在资源被虚拟机轮流共享时,除发生数据泄露外,拒绝服务攻击也将是家常便饭。

(17) 在可能的情况下,保证网络接口卡专用于每一个虚拟机。这里再次减轻了资源共享问题,并且虚拟机的通信也得到了隔离。

(18) 投资购买可满足特定目的并且支持虚拟机的硬件。不支持虚拟机的硬件会产生潜在的安全问题。

(19) 分区可产生磁盘边界,它可用于分离每一个虚拟机并可在其专用的分区上保障安全性。如果一个虚拟机超出了正常的限制,专用分区会限制它对其他虚拟机的影响。

(20) 要保证如果不需要互联,虚拟机不能彼此连接。前面已经说过网络隔离的重要性。要进行虚拟机之间的通信,可以使用一个在不同网络地址上的独立网络接口卡,这比将虚拟机之间的通信直接推向暴露的网络要安全得多。

(21) NAC 正走向虚拟机,对于基于虚拟机服务器的设备尤其如此。如果这是一种可以启用的特性,那么,正确的实施 NAC 将会带来更长远的安全性。

(22) 严格管理对虚拟机特别是对主机的远程访问可以使暴露的可能性更少。

(23) 主机代表着单个失效点,备份和连续性要求可以有助于减少这种风险。

(24) 避免共享 IP 地址,这又是一个共享资源而造成问题和漏洞的典型实例。

业界已经开始认识到,虚拟化安全并不是像人们看待物理安全那样简单。这项技术带来了新的需要面临的挑战。虚拟化安全是一项必需的投资。如果一个单位觉得其成本太高,那么建议最好不要采用虚拟化,可坚持使用物理机器,但后者也需要安全保障。

7.5　虚拟化未来发展趋势

在当今高度自动化、技术驱动的经济环境下，业务能力与 IT 能力正越发密不可分。IT 能力已经成为企业推动自身业务向前发展的重要赋能器，IT 基础设施从根本上决定了业务的成败。

然而，由于 IT 基础架构日益复杂，不灵活、脆弱和昂贵正成为其代名词，企业 IT 服务的成本由此水涨船高，导致投资预算穷于应付 IT 维护，无法最大限度地支持业务。相关调查显示，企业 70%以上的 IT 预算被运用于基础架构和应用程序维护以维持现状；仅有不到 30%的 IT 预算被运用于基础架构和应用程序的投资，以实现竞争优势和支持企业创新。

对那些希望通过 IT 能力拥抱业务创新梦想的企业而言，他们正迫切寻找一种新的 IT 服务模式，将应用和信息从底层基础架构的复杂性中解脱出来。事实上，这正是当今 IT 行业面临的最大挑战。

7.5.1　被重构的 IT 图景

类似的挑战在一个世纪前人类使用电力的进程演变中也曾出现。

当时，成千上万的工厂都拥有自己的发电机，这些发电机能解决自身的能源动力问题。私人发电设施的存在，增加了工厂主的固定资产投资，导致可用资本无法运用于直接推动业务增长的领域。同时，一旦技术过时或设备出现故障，便意味着庞大的更新及维护费用。

此后，科技和工程的一系列进步使具备中央形态的大型电厂开始出现。大型电厂集中生产的规模经济效应，促使电力成本大幅降低，效率急剧提升，使电力行业逐渐成为一种公用事业。对工厂主而言，他们不必再自建发电设施，转而从更加高效的大型电厂购买廉价电力。

如今，电力生产在一个世纪前遭遇的变革正在 IT 行业酝酿。由单个公司生产和运营 IT 系统的状况正在被中央数据处理工厂通过公共网络取代。IT 正在逐渐演变成公用设施，完成从工具到效用的转变，并由此诞生了一种崭新的 IT 服务模式，这便是云计算。

云计算是一种全新的运算方式，IT 基础架构可以作为一种可靠的、可随处获取的公用设施服务向用户提供，其概念类似于电力、电话。全球技术研究和咨询公司 Gartner 对云计算的描述是“一种新的运算方式，将可扩展的、富有弹性的 IT 功能作为一种服务提供，获得更好的技术共享(尤其是多租户应用或虚拟机)，增强的自动化，快速动态的改变，政策驱动及面向服务，减少的复杂性以及自动响应”。

云计算能有效降低成本、增加灵活性和提升服务质量，将应用和信息从底层基础架构的复杂性中解脱出来，使 IT 能够专注于支持和提升业务价值。

7.5.2　云计算的演进

基于硬件和软件基础架构的传统计算模式，让位于基于应用和服务提供的云计算模式是一个必然经历的过程。但是，由于商业、技术、管理等多个层面的挑战，该转变过程尚需经历三个重要的阶段。

首先,需要将现有的数据中心转化为内部云(Internal Cloud)。内部云服务是一种在内部IT环境中提供的云基础架构,它为内部的IT服务提供者带来了云计算的好处。例如,一些大型企业在全球往往有多个大型的数据中心,每个数据中心进行数据处理的繁忙时段并不相同。在如今网络传输速度早已不是问题的情况下,一旦企业认为自己所处地理位置的数据中心无法满足当前数据处理要求时,便可将任务远程交付于此刻处于闲置状态的其他数据中心。这便是内部云服务。

其次,当内部云服务出现之后,除单个企业内部的有效运用之外,同一行业或具备相同特性的对等企业之间也可实现数据处理服务的交换,这就是外部云(External Cloud)服务。

外部云服务是一种由托管或SaaS提供商等外部IT服务提供商提供并管理的云基础架构。举例来说,当前有许多企业一方面不希望无穷尽地进行IT投资,另一方面其实际的IT需求正有增无减。面对这样的挑战,企业之间可以通过相关手段利用对方的资源。例如,目前以银行为代表的众多机构,运算资源每年只为满足短短几天的峰值运算要求,其他时间段则处于闲置状态。假如其他企业希望在不影响银行本身的管理和安全机制的情况下租用闲置的运算资源为其服务,这便是内部云服务。当然,对于未来的外部云IT提供商而言,能否建立完善的管理机制、安全机制、费用清算机制以及具备足够的公信力,将是外部云能否尽早实现的决定因素。

最后,是公共云(Public Cloud)服务。该服务通常由第三方服务提供商通过公共或开放的网络向用户提供。就中国市场而言,最终扮演公共云提供商角色的厂商或机构,必然会与拥有网络资源的电信运营商产生某种程度的结合。电信运营商将会在涉及安全、计费等领域与包括虚拟化解决方案提供商在内的IT厂商紧密协作,共同构建公共云。

在早期,业界更多地将目光投向了外部的公共云服务,企图完全利用新的应用满足业务需求。然而,现实是残酷的,很少企业会因为新的架构而抛弃现有应用。实践证明,只有遵循渐进的、非破坏性的发展路径才能使“云”真正落地。

面对这一挑战,全球桌面到数据中心虚拟化解决方案厂商采取了更为注重实效的方法。首先,将现有的数据中心转化为内部云。同时,与托管和服务提供商合作,共同实现可兼容的外部云。随后,通过在云之间进行联邦和统一管理,使内部资源和可利用的外部资源链接起来,帮助企业获得云计算的所有好处和灵活性,这一结果实质上就是“虚拟私有云”。

作为一种跨越内部云和外部云基础架构的云计算环境,虚拟私有云为业务提供无缝的、可管理的云。这一概念类似于目前的虚拟专用网络(VPN)。如图7-22所示,虚拟专用网络是为适应业务需求,通过连接局域网(LAN)与广域网(WAN)资源,提供跨地域的、高效的

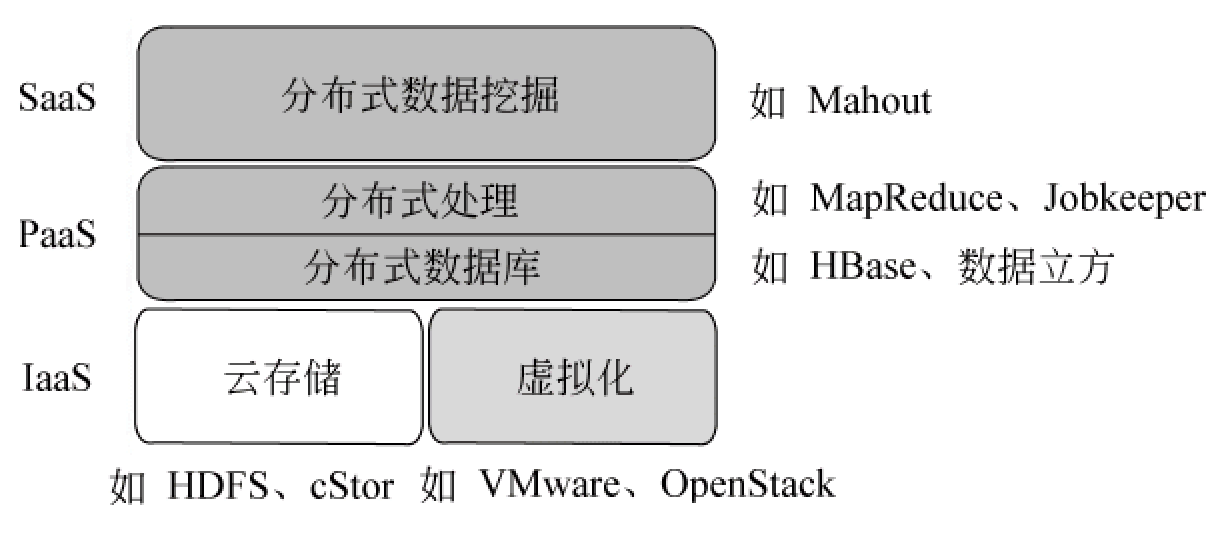


图7-22　架构示意图

网络访问和连接。通过利用内部和外部的公共基础架构，虚拟专用网络提供了极好的成本效益。此外，该业务通过无缝地接入网络，控制整个网络的接入和安全。这些概念同样适用于虚拟私有云。内部资源和外部可利用资源的结合，最大限度地提高了成本效益，并且保持了对整体 IT 基础架构的控制。与虚拟专用网络可跨网络服务提供商运行类似，虚拟私有云也可以跨服务提供商，从而确保灵活性和选择性。

虚拟私有云原则上有两个基本前提：第一，基础架构实现 100％的虚拟化，包括处理器、存储、网络等。第二，在可管理的服务水平协议(SLA)下，它能够跨越公共的、可利用的外部基础架构与可操作的内部基础架构。

虚拟私有云在集团 IT 控制之下带来了云计算的一系列好处。

首先，基于虚拟私有云，需要推出新业务服务的应用提供者，能够不受服务、存储和网络等基础架构复杂性的影响，专注于提供商业价值。

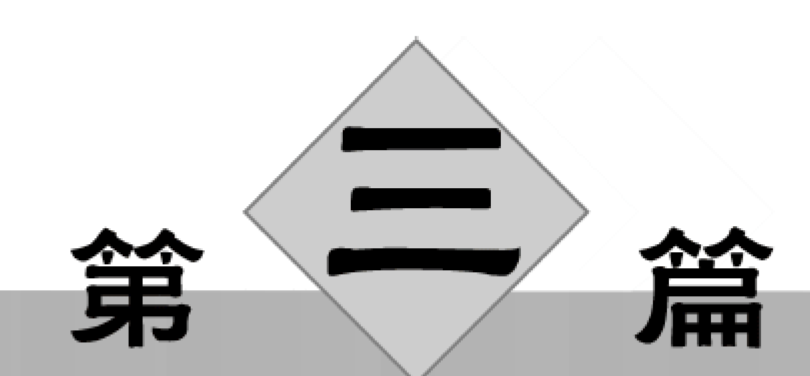
其次，更好的成本效益。这意味着去除了不必要的投资，获得更低的总拥有成本并且使花费在管理和维护 IT 基础架构上的时间和投资最小化。用商业术语来说，成本效益就是指极大地节省投资成本和运营成本。利用 VMware 领先的虚拟化解决方案以及对跨内部云和外部云基础架构的资源进行自动化管理和动态优化，VMware vSphere 4 实现了最高的整合率。这一结果最直接的影响就是突破了昂贵的应用和信息提供模式，将传统的、依赖于特定系统和架构的应用和信息提供模式转变成自管理、动态优化的环境，从而最有效地提供业务服务。

最后，是宽泛的选择性。利用 VMware vSphere 4，客户可以保持选择的灵活性，继续独立于硬件、操作系统、应用栈和服务提供商。这就意味着客户既可以支持他们现有的应用，又可以对将来的应用部署有充分的信心，灵活地在内部云或外部云上部署应用。

任务拓展

1. 什么是虚拟化？
2. 请简单介绍四种虚拟化架构。
3. ESX 虚拟化架构有哪些优劣势？
4. 虚拟化技术为什么可以作为云技术的基础？

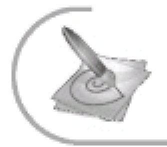




数据灾备应用

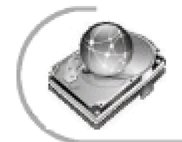


第 8 章　灾备系统设计与典型案例分析



任务目标

- 了解硬件灾备架构；
- 了解数据灾备架构；
- 了解成本控制与风险控制；
- 学习典型灾备案例。



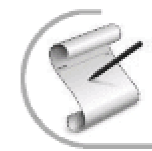
项目背景

现今无论对于大型企业还是其他组织，都拥有为数众多的服务器，而其中的数据则是最为重要的财产。系统故障的出现，可能导致企业生产停顿、政府部门无法正常履行职能等严重后果。因此，保持业务的持续性是用户在选择计算机系统、部署数据存储的重要指标。究其根本原因，保护业务持续性的重要手段就是提高计算机系统的高可靠性，同时将数据的损失降至最低限度。

RTO/RPO 是数据灾备系统的两个非常重要的指标，所有灾备系统搭建都是以 RTO/RPO 为目标来进行定制，但是 RTO/RPO 值越小，其灾备系统面临的架构会越复杂，建设系统的成本会越昂贵。作为项目实施方案的规划者，必须充分了解系统自身架构，充分理解系统的实际用途，并且结合自身的合理判断，来提供最切合实际的解决方案。

本章介绍硬件和数据灾备架构，以及成本控制与风险分析，并列举典型案例，介绍通过实际系统需求分析来为项目搭配合适的应用及数据灾备架构。以此让大家进一步加深对灾备系统在现实生活中的作用。

需要特别注意的是，本章典型案例的介绍和示意图均以应用和数据灾备功能描述为主，案例将对系统本身应用及硬件架构进行简化，以方便大家的理解。实际应用中，大多数机构的业务系统架构会比本章案例更复杂。



项目描述

聚比特科技有限公司已经构建了自己的灾备系统，为了提高自身灾备系统的可用性，加强对灾备系统的掌握，公司要求技术部门对灾备系统进行各种场景下的模拟和测试，明确优势，避免劣势，增强灾备系统的实际运用水平。



项目分析

聚比特科技有限公司技术部针对自身灾备系统进行认真分析，结合灾备系统的两大指标

RTO/RPO,发现了自身灾备系统的优势和缺点,决定对公司灾备系统进行一次大的改造升级。



项目实现

聚比特科技有限公司技术部认真研讨改进和升级设计思路,明确灾备系统设计原则,模拟公司可能遇到的数据安全威胁,并参考各典型案例,为灾备系统的升级定下了明确的目标和实施办法。

8.1.1 信息可行性分析

全面详尽的风险分析是数据中心灾备体系设计的基础,风险分析方法包括以下方面。

(1) 资产识别。主要包括基础设施、硬件、软件、数据、文档、服务和声誉等。单位应对资产进行分类,以区分资产的不同重要程度并确定重要资产的范围,应对资产进行标识以区分资产对业务正常运作的影响程度,据此确定资产的等级。

(2) 威胁识别。即识别信息资产构成潜在破坏的可能性因素,如自然因素与人为因素、内部因素与外部因素等。

(3) 脆弱性识别。即识别可能被威胁利用的信息资产的弱点,主要包括技术与管理两个方面。技术脆弱性涉及物理层、网络层、系统层、应用层等各个层面的安全问题;管理脆弱性可分为技术管理脆弱性和组织管理脆弱性两方面,前者与具体技术活动相关,后者与管理环境相关。

具体分析活动可通过问卷调查、工具检测、人工核查、文档查阅和渗透性测试等方式开展。完成风险分析后,需要根据灾难发生的可能性、灾难发生后的损失预计等因素,计算对应的风险值,进行风险分级,为后续分析工作提供参考。

8.1.2 故障分析

1. 定义

业务影响分析(Business Impact Analysis,BIA)的目的是确定不同业务遭遇到风险后对企业的影响程度。通过分析有形的和无形的影响,估算对停止业务时间长短的接受情况和使影响降至最低的处理需求,对灾备的具体实现提出明确要求。BIA 的目标包括以下内容。

(1) 识别和量化每个业务单元或者资源对整个企业在业务运行方面的影响。

(2) 识别潜在的失效场景和评估潜在的威胁。

(3) 定义针对不同的灾难恢复要求所需要的不同级别的投资情况。

(4) 建立灾难恢复时的恢复流程优先级,指导灾难备份恢复策略的制定。

2. 业务影响分析的方法

业务功能地位分析。可从业务功能的政策要求、是否为核心业务、业务涉及的机构与用户范围、业务处理实时性与频度、业务功能与机构内外其他业务功能的关联等多个角度综合分析。

业务中断影响评估主要包括两个方面,一是以量化的方法,评估业务中断可能带来的直接与间接经济损失;二是以非量化的方法,评估业务中断所引发的社会影响、法律影响、信用影响、品牌影响等。

业务影响分析最终将影响对灾难备份体系的需求,根据规范的要求,灾备体系的需求必须明确需求等级、各等级对应的最低恢复要求以及恢复的优先级。主要指标有 RTO (Recover Time Objective),即灾难发生后,信息系统从停顿到恢复正常的时间要求;以及 RPO(Recover Point Objective),即灾难发生后,数据必须恢复到正常状态的时间要求。灾难恢复能力等级与上述两项指标的参照要求如表 8-1 所示。

表 8-1　灾难恢复能力等级及 RTO、RPO 指标要求

灾难恢复能力等级	RTO	RPO
1	2 天以上	1～7 天
2	24 小时以上	1～7 天
3	12 小时以上	数小时至 1 天
4	数小时至 2 天	数小时至 1 天
5	数分钟至 2 天	0～30 分钟
6	数分钟	无限趋近于 0

8.1.3　基础架构分析

数据中心技术体系分析是灾备技术体系建设的基础,灾备技术体系与数据技术体系匹配程度的高低,决定了灾备中心对数据中心生产环境的可替代程度,对于高效实现灾难恢复,提高灾备中心可用性有重要意义。对数据中心技术体系的分析主要从以下两方面入手。

(1) 基础运行环境分析。在确保灾备中心与数据中心技术架构基本一致的前提下,明确所需要的设备类型和数量,对机房配电、空调、地板承重以及布线的具体要求等基础环境信息,为选择具体的灾备环境提供参考。

(2) 应用部署特点分析。主要分析各类应用部署平台的情况、应用之间的数据依赖关系、应用正常运行需要数据质量、应用正常启动和异常启动需要的时间等关键因素,作为灾备中心应用体系构建的基础。

例如,某集团数据中心,生产拓扑图如图 8-1 所示。根据用户的现有情况,用户提出了以下需求。

1. 数据备份

需求:需要对现有网站服务器文件数据、ERP 服务器文件数据以及数据库数据进行定时备份。

分析:用户目前的网络架构清晰,设施设备运行情况良好,网络环境优越且有足够的网络带宽支撑灾备系统建设,系统管理人员相对紧张,技术薄弱。

建议:备份服务器+备份软件+备份介质。采用存储备份数据,生产机实时向备份发送关键数据,如图 8-2 所示。

2. 云桌面备份

需求:需要对虚拟化桌面平台进行虚拟桌面备份。

分析:虚拟化桌面平台对外服务,员工使用虚拟桌面创建数据,此类平台特点为虚拟桌面多,数据类型不一,每个人操作习惯、时间、文件类型不同。

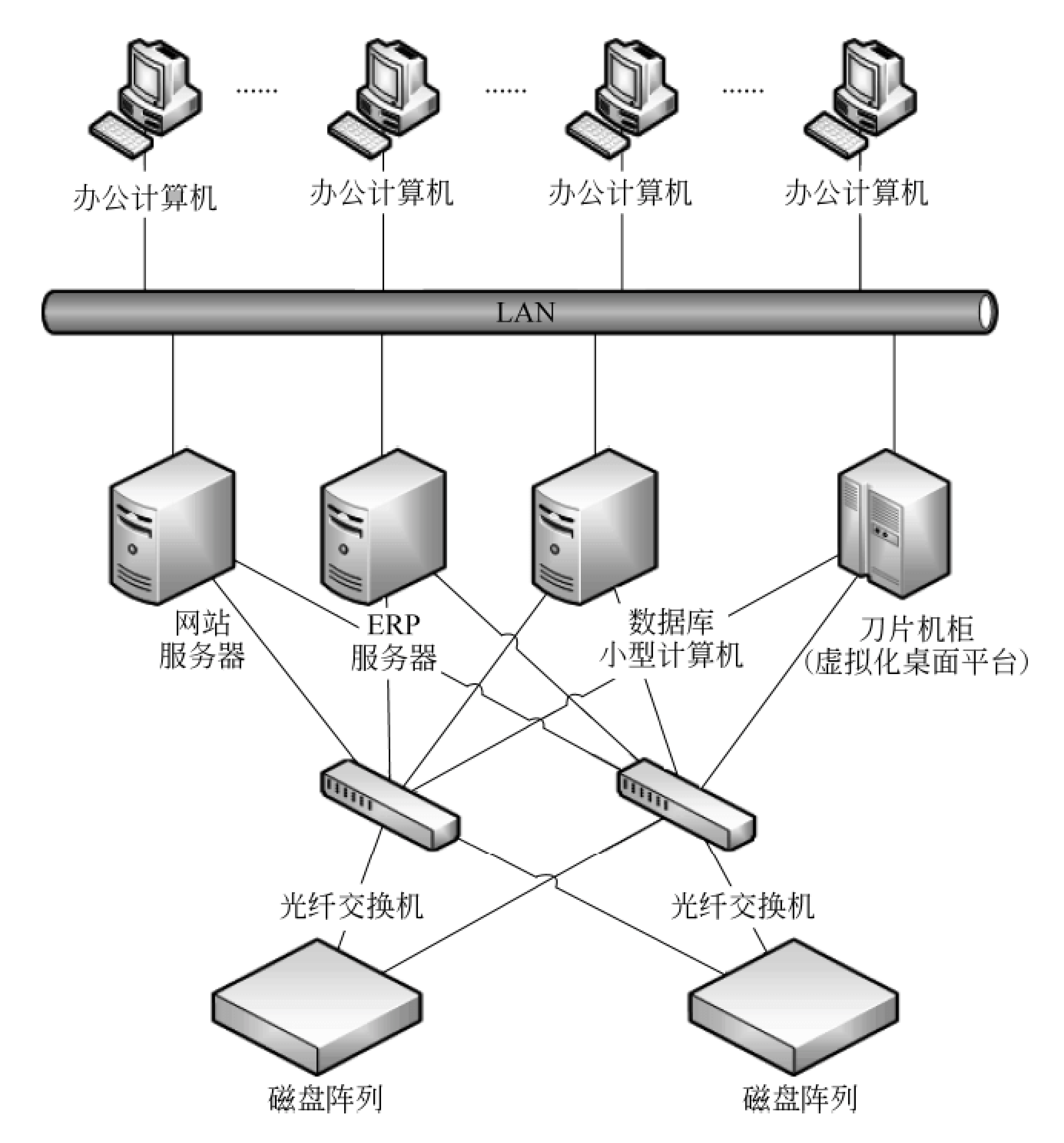


图 8-1　某集团数据中心生产拓扑图

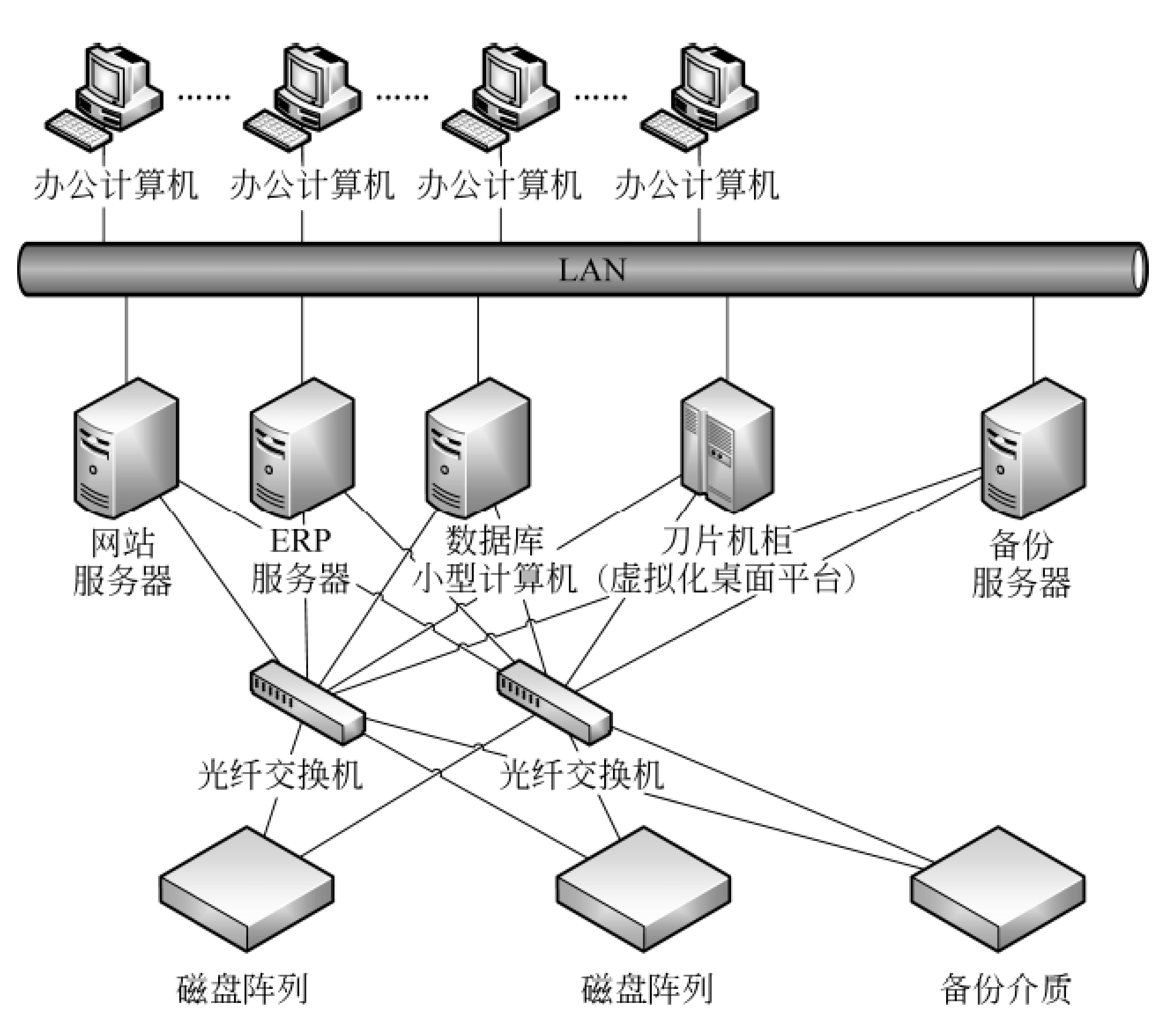


图 8-2　某集团生产中心灾备方案拓扑图

建议：采用云桌面备份方式，常见的云桌面备份方式步骤如下。

(1) 检测目标 PC 开机，查看记录，了解该 PC 今日是否已完成备份，开启备份任务。

(2) 备份过程中该 PC 关机，备份任务停止并记录。

(3) 该 PC 开机,发现有备份任务未完成,继续完成该备份任务。

(4) 检测到 PC 开机,但记录中显示该 PC 今日已做过备份,所以不触发备份任务。

3. 业务接管

需求:网站服务器支撑公司官网系统,数据库服务器中由 Oracle 数据库提供服务,用户需要让服务器实时对外提供服务。

分析:此类系统部署简单,需要有一套装有相同应用系统的硬件支撑。

建议:双机软件+服务器,当生产系统出现物理故障,会有相同的一套系统接管业务,实时对外服务。如图 8-3 所示。

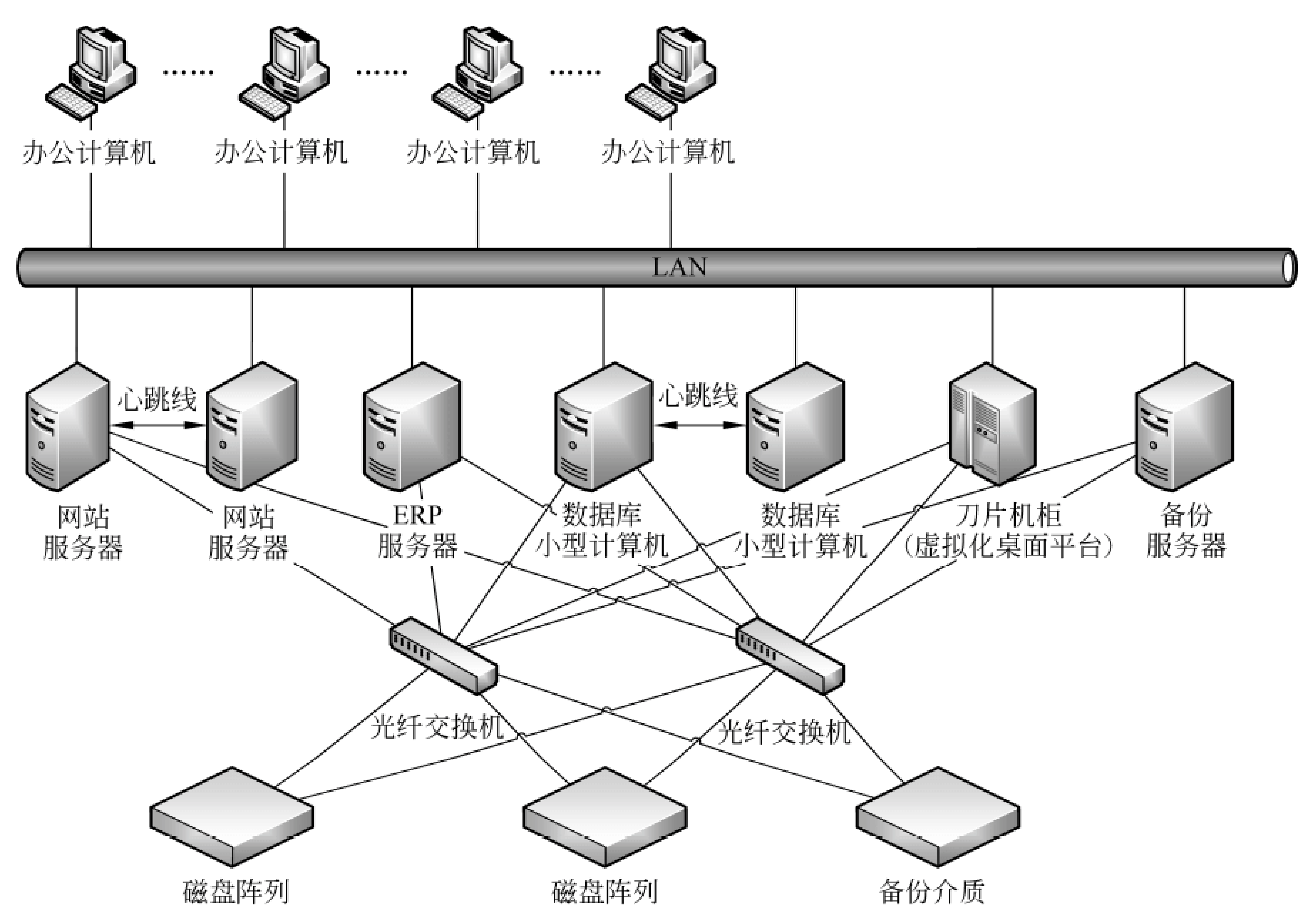


图 8-3　某集团生产中心云桌面备份拓扑图

4. 灾备中心建设方案

需求:随着业务发展的壮大,用户提出了更高的灾备需求,需要在本地(另一大楼)建立一灾备机房,并将 ERP 服务器实时接管到灾备机房且做到数据的实时复制。

分析:异地机房建设,可保证生产机房发生故障时能不影响对外业务,此灾备方案既能做到异地灾备且投入相对较小。

建议:采用 CDP 技术,可保证 IP 可达,能实时接管业务并对外提供服务,如图 8-4 所示。

5. 双活数据中心

需求:因成本考虑,用户希望灾备机房设备能对外提供服务,减轻生产系统的压力。

分析:该需求主要考虑的是同步数据两端都需要数据是实时可用的,是互为同步的,这样才能保证两中心均可对外提供业务。

建议:在本地机房建立灾备中心,部署双活数据中心,灾备端实时对外提供服务,减轻生产系统压力,如图 8-5 所示。

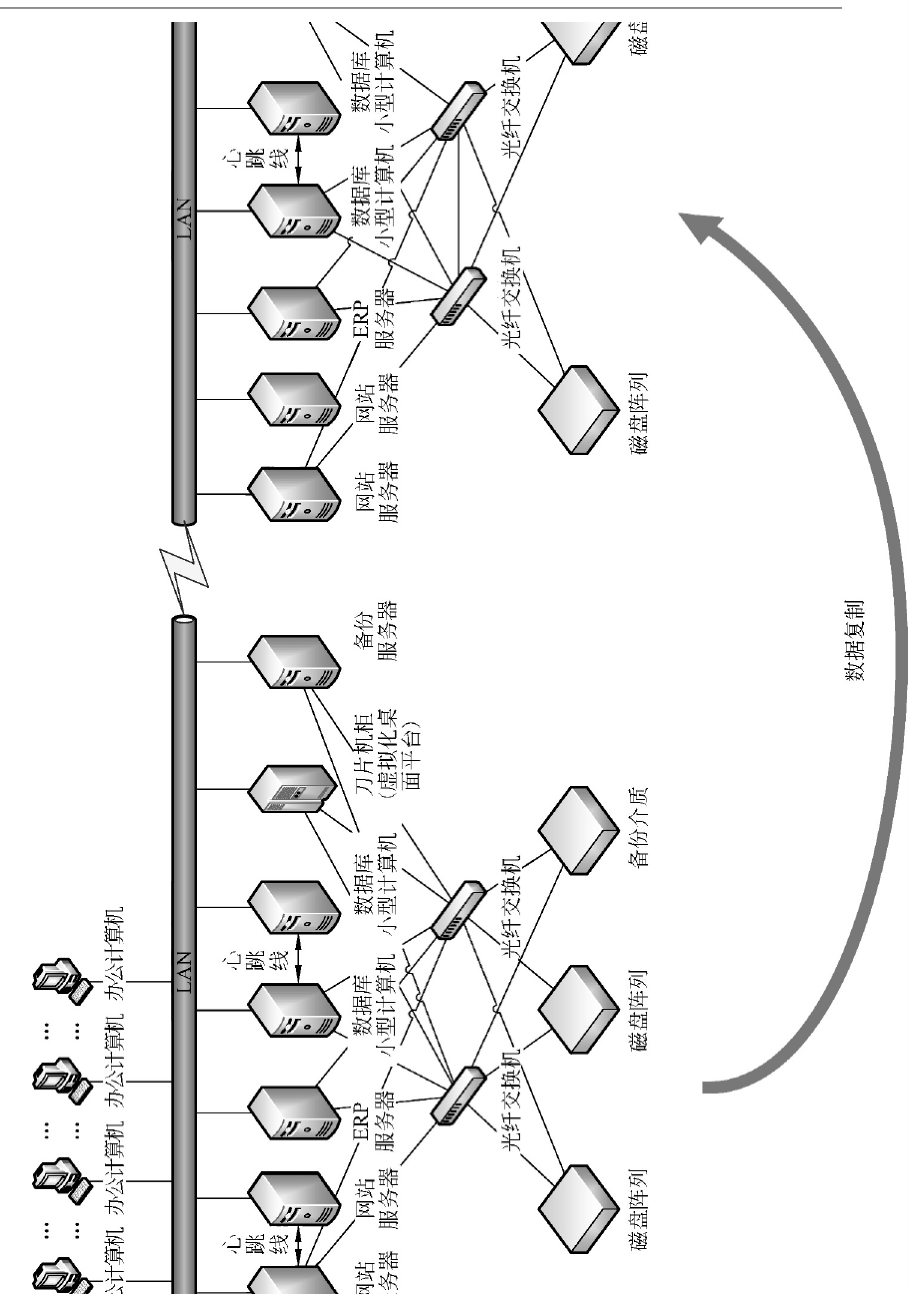


图 8-4　某集团生产中心灾备中心建设拓扑图

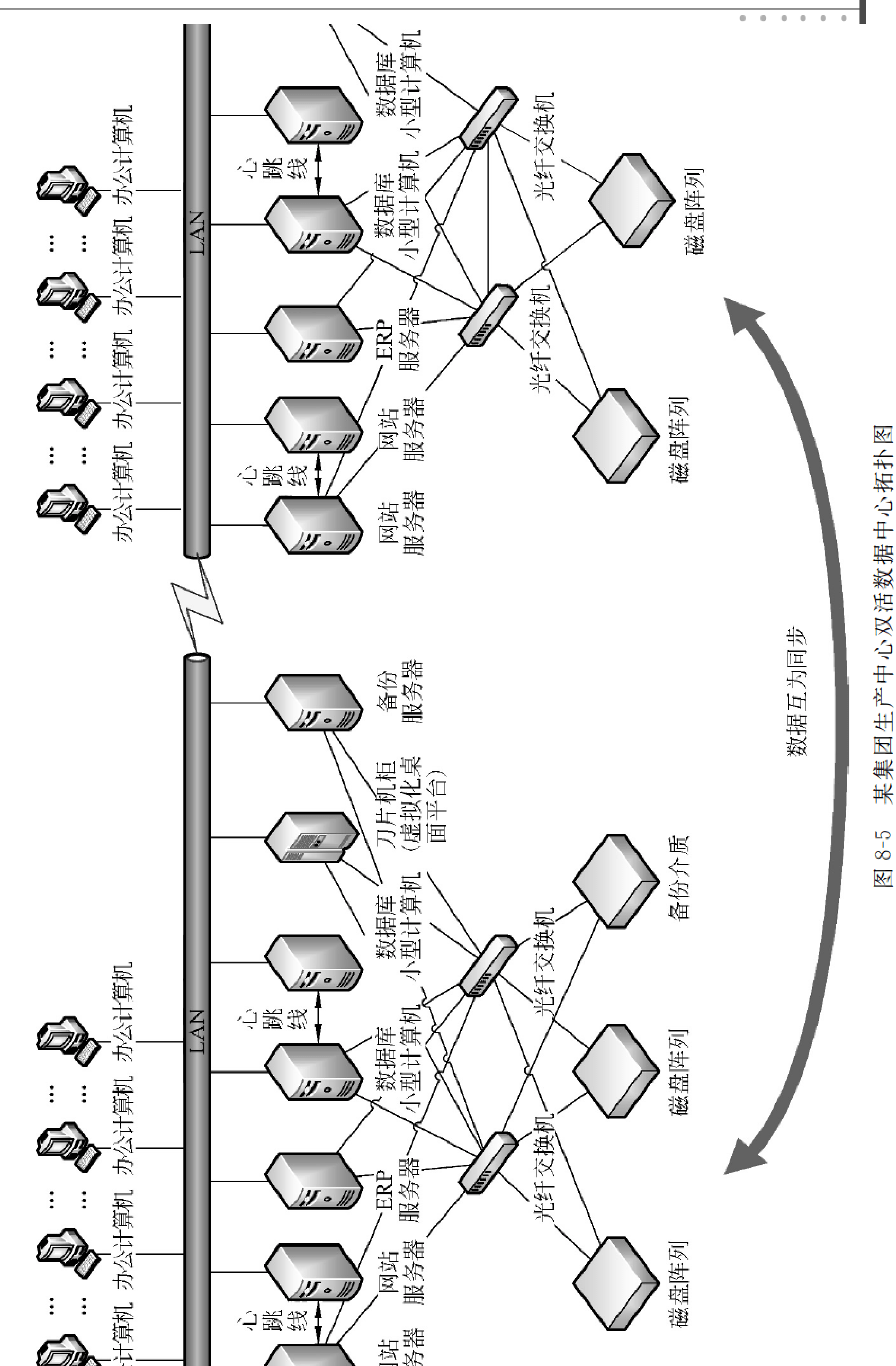


图 8-5 某集团生产中心双活数据中心拓扑图

6. 两地三中心灾备

需求：为防止本地地震、大面积停电、战争等可能导致本地两机房同时无法运行的情况，需要在异地建立灾备中心。

分析：该架构就是灾备中心和双活中心的结合体，既有双活中心的冗余系统架构，也有异地的灾备效果，如图 8-6 所示。

建议：远程异地建设灾备中心，防止大面积灾难事故造成的影响。

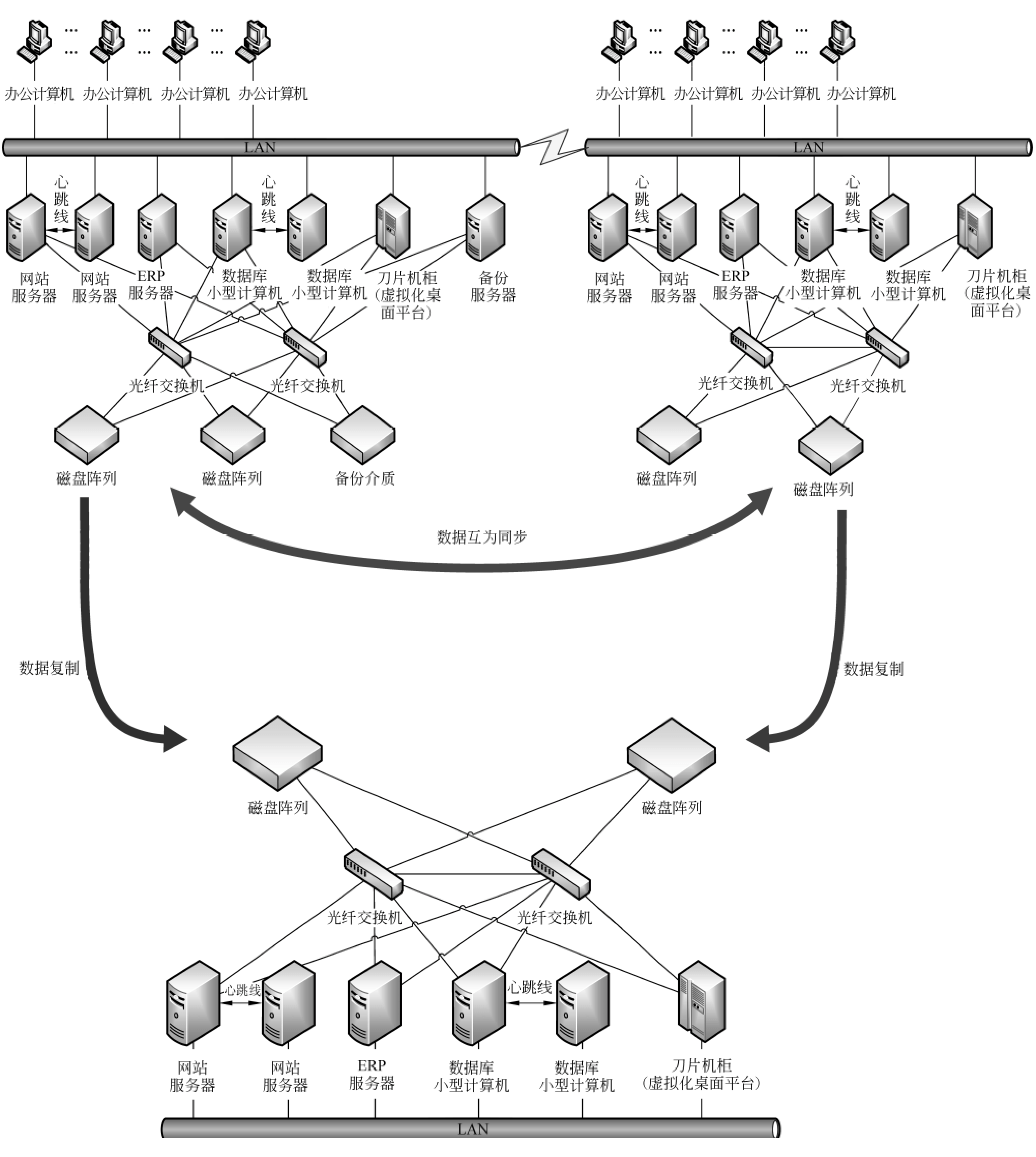


图 8-6　某集团生产中心两地三中心拓扑图

8.2　系统的设计思路和设计原则

8.2.1　系统设计的理论依据和规范

本技术方案将参照以下标准和规范：

(1)《信息系统灾难恢复规范》(GB/T 20988—2007)；

(2)《计算机场地安全要求》(GB/T 9361—2000)；

(3)《信息技术 信息安全管理实用规则》(GB/T 19716—2005)；

(4) 美国国家标准和技术学会推荐的《信息系统应急计划指南》。

8.2.2　系统设计方法论

政务数据资源中心的建设将通过对成都市信息办各应用系统的需求及各委办局应用系统及 IT 基础环境的调研，结合政务数据资源中心 IT 资源的实际情况，制定出符合不同应用系统特点的数据存储及备份策略，根据不同的数据存储及备份策略，采用相应的技术和产品，并提出数据资源备份中心系统运行和维护方案，具体方法如下。

(1) 现有系统资源调查：通过标准的系统资源调查问卷，了解信息办生产系统及各单位 IT 基础环境情况(包括 IT 基础设施情况、应用系统情况、网络连接架构等)。

(2) 数据备份需求调查：通过与成都市信息办及各部委办的访谈，了解各单位数据容量及备份的需求，包括数据类型、数据容量和分布情况、数据备份的频率和数据恢复要求。

(3) 数据备份需求分析：对采集的各单位数据进行分析。

(4) 制定数据备份的策略：根据需求分析结论，进行数据容量规划和制定数据备份策略，采取相应的产品和技术手段。

(5) 制定政务数据资源中心建设方案：方案的内容包括对政务数据资源中心的 IT 资源分析，备份数据资源整合建议，数据备份与恢复建议、运维管理建议、专业化服务建议。

(6) 制定政务数据资源中心的实施方案：实施方案主要包括系统实施计划、实施内容、实施策略等。

8.2.3　技术路线

1. 系统可用性

系统可用性机制是系统安全平稳运行的可靠保证，在政务数据资源中心的存储设备和备份设备中要提供相应的数据保护机制和数据验证机制，数据保护机制是指对政务数据资源中心的数据提供 RAID 级数据保护，并根据不同数据类型提供不同的 RAID 级别，以提高数据的安全性和可靠性；数据验证机制是指定期对数据的完整性、一致性和可用性进行验证，保证备份数据的可用性。

2. 抗灾能力

按照灾难恢复和业务连续性体系建设的相关规范，制定切实可行的灾难恢复策略和灾难恢复预案，并根据应用系统的特点进行定期演练，提高系统抵御灾难的能力。

3. 技术支持能力

技术支持能力主要体现在对外技术支持能力和对内技术支持能力两个方面，其中对外

支持能力主要体现在对各委办局提供的后台技术支持，包括数据备份与恢复服务、数据测试服务、应急响应预案及演练、数据验证服务和培训服务等。对内支持能力主要体现在对政务数据资源中心内部 IT 基础设施的支持能力，包括政务数据资源中心自身生产系统的运行和数据存储实施及测试服务、本地数据备份服务、培训服务等。

4. 运维管理能力

系统运维管理能力主要体现在对政务数据资源中心各种资源的运维管理能力，由于这些资源在集成运行中的不确定性，系统故障随时都有可能发生，因此需要建立一整套系统运行维护管理体系来保证系统的稳定运行，本项目的系统运维管理体系建设主要包括 IT 系统监控、运行维护管理的制度和流程、运维管理人员组织架构和岗位职责等。

5. 系统可恢复能力

系统的可恢复能力主要包括数据的可恢复能力和应用系统的可恢复能力两个方面，本期主要考虑数据的可恢复能力。数据的可恢复能力主要是通过对数据完整性和一致性检验手段，可以保证政务数据资源中心数据的可用性，一旦生产数据出现故障，备份的数据应能够回传到生产中心并恢复，保证生产业务系统的正常运行。

8.2.4 系统设计原则

1. 标准性原则

由于各应用系统的 IT 设备、技术手段和数据格式有所不同，要想建立统一的数据备份平台和应用平台，就必须依照国际上的规范和标准，并采用符合国际标准的接口、规范和协议，以保证各种异构平台之间的无缝连接。

2. 安全性和可靠性原则

为数据和应用的安全性和可靠性，应该从技术手段、备份策略、软硬件产品等多方面对安全性和可靠性做出保证。为了保证数据安全，除了建立可靠数据备份系统之外，还要采用成熟的技术和成熟的软硬件产品。

3. 先进性和成熟性原则

数据中心对主机设备、网络设备和存储设备的性能、容量、吞吐能力及稳定性、安全性要求很高，在设计过程中，要充分借鉴国内外流行的主流产品和技术，使用成熟的模式和先进的备份技术以及业界领先的产品，才能使备份系统不断地保持其先进性和技术的延续性。

4. 实用性原则

在建立数据备份产品的选择上应充分考虑产品的性能价格比，在确保功能实现和性能稳定的基础上，采用实用性高的产品，同时还要考虑对现有资源的保护。

5. 灵活性和扩展性原则

数据中心的服务是一个长期持续的过程，随着生产系统的增加及更多的系统接入，数据存储和备份需求将逐步增长，因此数据备份系统应具有良好的扩充能力和灵活的体系架构，以满足现在乃至未来较长一段时间内的数据容量和用户数量增长的需求，并具备支持未来关键系统应用级灾备的能力。

6. 可维护性和可管理性原则

数据中心的系统涉及多种 IT 系统，为了保证系统良好运转，要求系统具有良好的可维护性和可管理性，采用功能强大的系统监控机制，实现对数据存储、备份、恢复的全过程监控

和故障告警。

8.3 备份系统建设的重要性

计算机系统可轻松进行复杂工作中巨大数据量的处理工作，却也同样因其保存有巨大数据量的信息而存在一些数据安全上的隐患。一旦因为这些隐患导致数据丢失，其造成的损失几乎无法通过人力来达到短时间恢复。

8.3.1 系统灾难分析

1. 灾难类型

(1) 自然灾害

造成计算机灾难的自然灾害有火灾、水灾、雷击、台风、地震、鼠害等。

(2) 计算机系统故障

引起计算机系统故障的因素有下述几点。

① 主机系统故障：主要指数据库系统故障、系统软件故障、硬盘损坏、网卡故障、电源故障、应用系统缺陷及其他故障。

② 主机房故障：主要指主机房电源故障、主机房通信故障、主机房水灾、主机房火灾及主机房鼠害。

③ 整幢楼房故障：主要指整幢楼房电源故障、整幢楼房火灾或水灾、整幢楼房其他灾害。

(3) 人为因素

应用系统缺陷、误操作、人为蓄意破坏、外来暴力事件等，都将直接影响系统的安全运行。

2. 系统灾难统计

图 8-7 所示为国内银行业信息系统灾难情况统计数据。

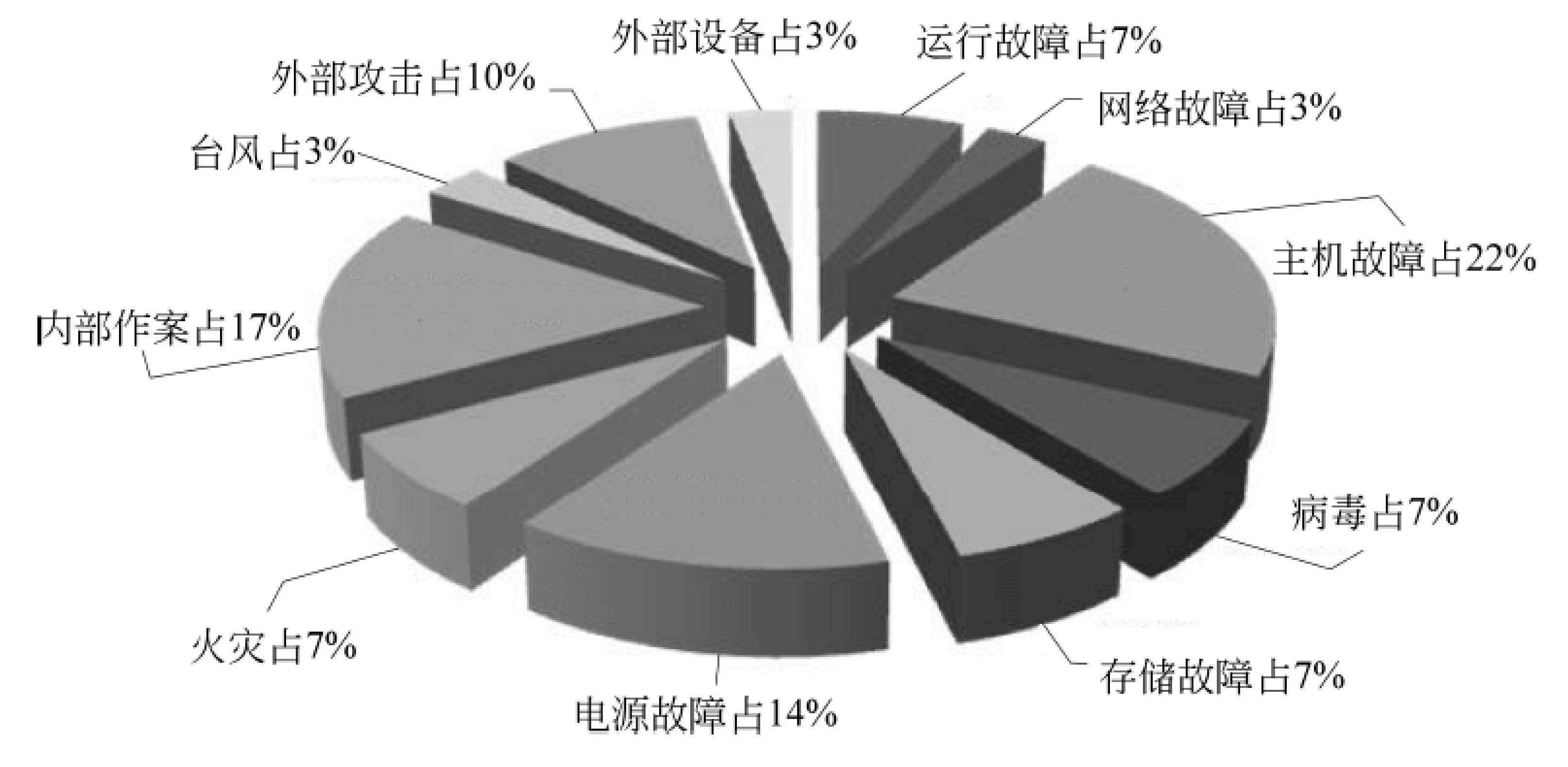


图 8-7　系统灾难统计图

8.3.2 国家对数据安全的重视

国家及监管部门对加强重要信息系统安全保障工作非常重视，先后出台了多项有关信息安全的意见和指导建议。其中部分管理规定及文件如表 8-2 所示。

表 8-2 信息系统安全保障的部分管理规定及文件表

发布部门	发布时间	名称
中共中央办公厅、国务院办公厅	2003	《国家信息化领导小组关于加强信息安全保障工作的意见》(中办发[2003]27 号)
国务院信息化工作办公室	2005	《重要信息系统灾难恢复指南》
国务院信息化工作办公室	2007 年 7 月	国家标准《信息系统灾难恢复规范》(GB/T 20988—2007)

8.3.3 备份系统保护数据安全

备份系统是一套针对应用连续性和数据安全性保护的系统，它通过统一化、自动化的操作对系统应用和数据进行有针对性的保护动作。

8.3.4 备份系统的保护场景

按照监管机构关于《信息系统灾难恢复规范》中重要信息系统应急预案的要求，应急场景应覆盖电力故障、通信线路故障、火情水灾、大规模区域性灾难、治安、病毒爆发、网络攻击、人为破坏、不可抗力、计算机硬件故障、操作系统故障、系统漏洞、应用系统故障以及其他各类与信息系统相关的故障。这些故障由于在爆发的诱因、破坏程度和造成的影响等方面都有所不同，因此应针对不同场景发生时制定相应的应急处置手段和策略。

8.4 典型案例分析

8.4.1 基本数据保护及数据归档场景

基本数据保护场景常用于机构系统可接受数据少量丢失，系统恢复时间一天以上的情况。而数据归档场景常用于机构系统数据需要长时间保存的情况。

以档案局为例，档案局的工作主要是负责接收、征集、整理地区党政机关的重要档案资料，收集散失在地区内外与地区有关的档案文件和史料，做好机关、企事业单位的档案管理工作，做好档案编研和利用工作。所以档案局每天都会进行大量的档案录入，并且按照国家规定，档案的保存时间一般都长达 20 年以上。

档案局在建立灾备系统的时候，就需要从安全性、可进行数据归档、后期维护难度以及建设成本进行考虑。

1. 档案局系统的特点

档案局系统的特点如下。

(1) 数据量非常大，而且根据类别不同，数据会存放在各种系统平台下，如 Windows、

Linux、UNIX 等。

(2) 数据可以接受短时间的数据丢失，一般来说是 24 小时内，因为有档案原件在，数据少量丢失可迅速重新录入。

(3) 普通系统可接受短时间停用，一般来说是 1～2 天，不会影响其主要工作。核心系统可通过虚拟化进行应用级保护。

(4) 数据需要进行长期归档，通常是将数据备份到物理磁带库中进行归档。

2. 可进行的分析

根据档案局的特点，可以进行以下分析。

(1) 数据存放在多种系统平台下，所以不能采用 CDP 的方式进行备份。

(2) 数据可以接受短时间的丢失，从成本考虑，可采用定时备份架构。

(3) 数据需要归档，但直接备份到物理磁带库中，会使得备份及还原时间大幅提高，可考虑采用 D2D2T 架构(Disk-to-Disk-to-Tape)，也就是先将数据备份到磁盘阵列或 VTL 中，再从后台慢慢将备份数据导入物理磁带库，以便归档。

如此，档案局的灾备架构就展现出来了，如图 8-8 所示。

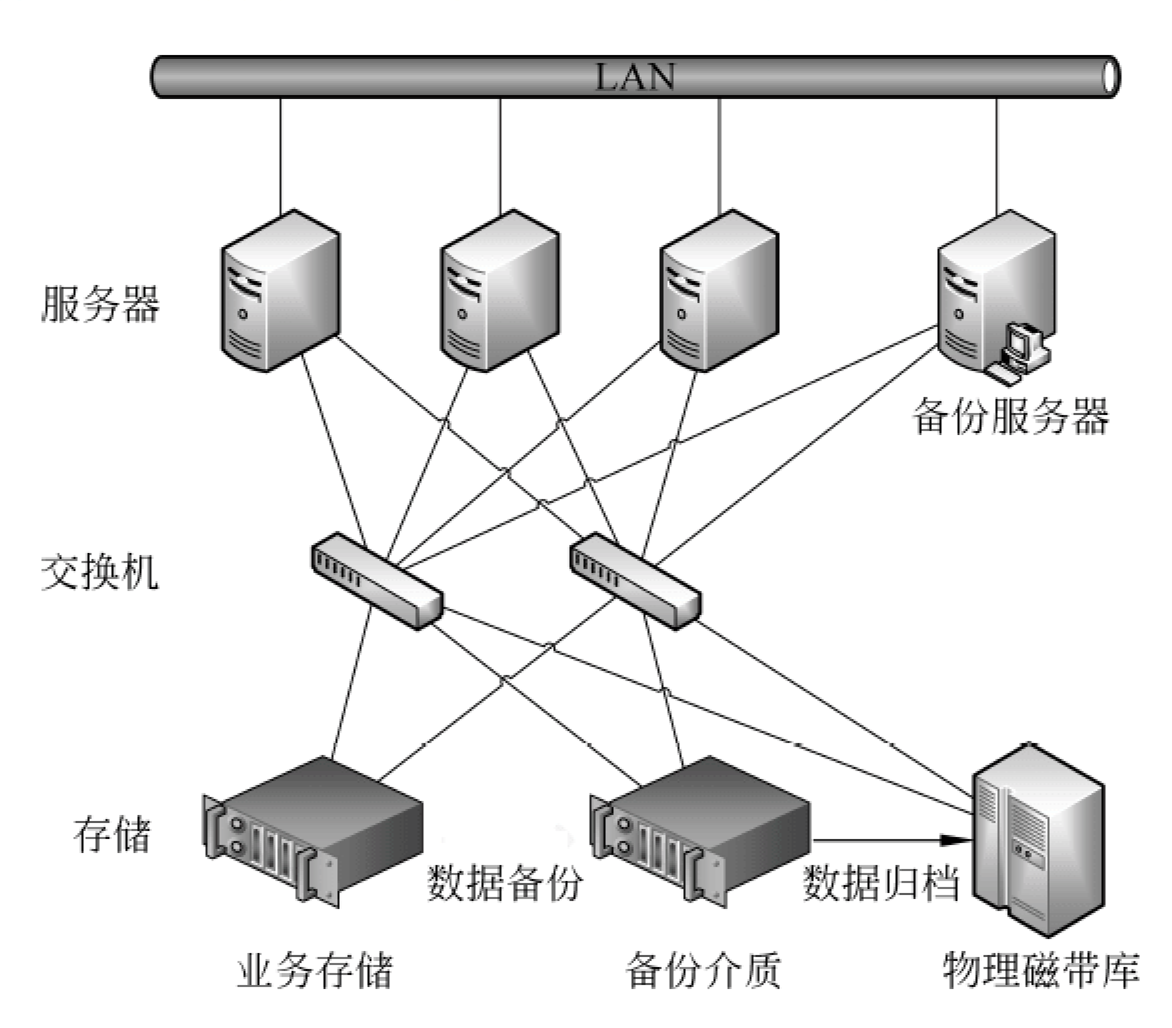


图 8-8　基本数据保护及数据归档场景拓扑图

上图中将备份服务器、备份存储、物理磁带库连入原有系统的 SAN 环境中，即可进行数据备份及数据归档工作。

3. 备份的过程

备份的过程如下。

(1) 备份服务器根据制定的策略，对每个服务器的客户端发出备份指令。

(2) 服务器上的备份客户端接收到备份服务器发送的指令后，快速定位需要备份的数据，并将数据在内存中进行加密后传输到备份介质中。

(3) 传输过程完成以后，备份系统还需要对备份介质中的数据进行可用性检查，检查通过后提交备份服务器，这时备份服务器会把该备份任务标记为"完成"。

(4) 备份任务完成后，备份服务器将启动数据归档任务，直接将备份介质里的数据传送到物理磁带库中进行归档保存。

4. 优势

该架构的优势表现在以下几方面。

(1)在数据发生丢失或者产生逻辑性故障时,备份服务器可以直接调用备份介质中的数据对业务数据进行还原。

(2) 备份数据转移到磁带库的动作可以直接通过高速的存储网络进行,而不用占用业务网络,并且不会增加前端的备份时间窗口。

8.4.2 应用保护场景

应用保护场景常见于机构部分关键应用不能长时间停机(RTO 值小)的场景。

以医院为例,医院系统平台是由多种系统构成的,主要由 HIS、LIS、PACS 等系统承担其主要功能。

医院信息系统(Hospital Information System,HIS)是覆盖医院所有业务和业务全过程的信息管理系统,其为医院所属各部门提供病人诊疗信息和行政管理信息的收集、存储、处理、提取和数据交换功能并满足授权用户的功能需求。可见一旦医院的 HIS 发生故障,将直接导致医院的各项工作无法开展。医院作为一个 24 小时对公众开放的社会机构,一旦无法工作,不仅将造成医院的直接经济损失,甚至可能增加社会不稳定因素。

医院检验系统(Laboratory Information Management System,LIMS)是 HIS 的一个重要的组成部分,其主要功能是将检验的实验仪器传出的检验数据经分析,生成检验报告,这样医生能够方便、及时地看到患者的检验结果,其主要依附于 HIS 进行统一管理。

影像归档和通信系统(Picture Archiving and Communication Systems,PACS)是应用在医院影像科室里的系统,主要的任务就是把日常产生的各种医学影像(包括核磁、CT、超声、各种 X 光机、各种红外仪及显微仪等设备产生的图像)通过各种接口以数字化的方式保存起来,当需要的时候能够很快地调回使用,同时辅助医生诊断管理。

可以罗列出医院主要系统的特点如下。

(1) HIS、LIS 的 RTO 值必须要低。

(2) PACS 的影像数据较多,这部分数据量大,同时 PACS 的 RTO 要求没有 HIS 那么小。

根据以上系统的特点,可以进行分析如下。

(1) HIS、LIS 要求 RTO 值低,所以需要考虑采用数据复制/镜像的方式进行保护。

(2) 数据复制/镜像无法解决数据的逻辑性故障,所以需要增加定时备份。

(3) PACS 数据量大,采用数据复制/镜像的方式会占用大量网络资源,可采用定时备份的方式,在医院业务较少的时候,比如凌晨以后再进行备份。

(4) 同时,在考虑应用结构的同时,还需要了解相关机构的政策条款。比如医院级别评审中明确对备份做了要求,要求必须有异地备份,所以在规划医院的灾备系统时,需要将异地灾备考虑在内。

综上所述,医院的灾备架构基本如图 8-9 所示。

由上图可见,异地灾备机房有一套备用的 HIS、LIS 服务器,通过数据复制技术,保证业务服务器和备用服务器的数据一致,一旦业务服务器发生故障,可马上让备用服务器将所有业务进行接管,保证业务的连续运行。同时,PACS 通过备份软件将数据备份到异地灾备机

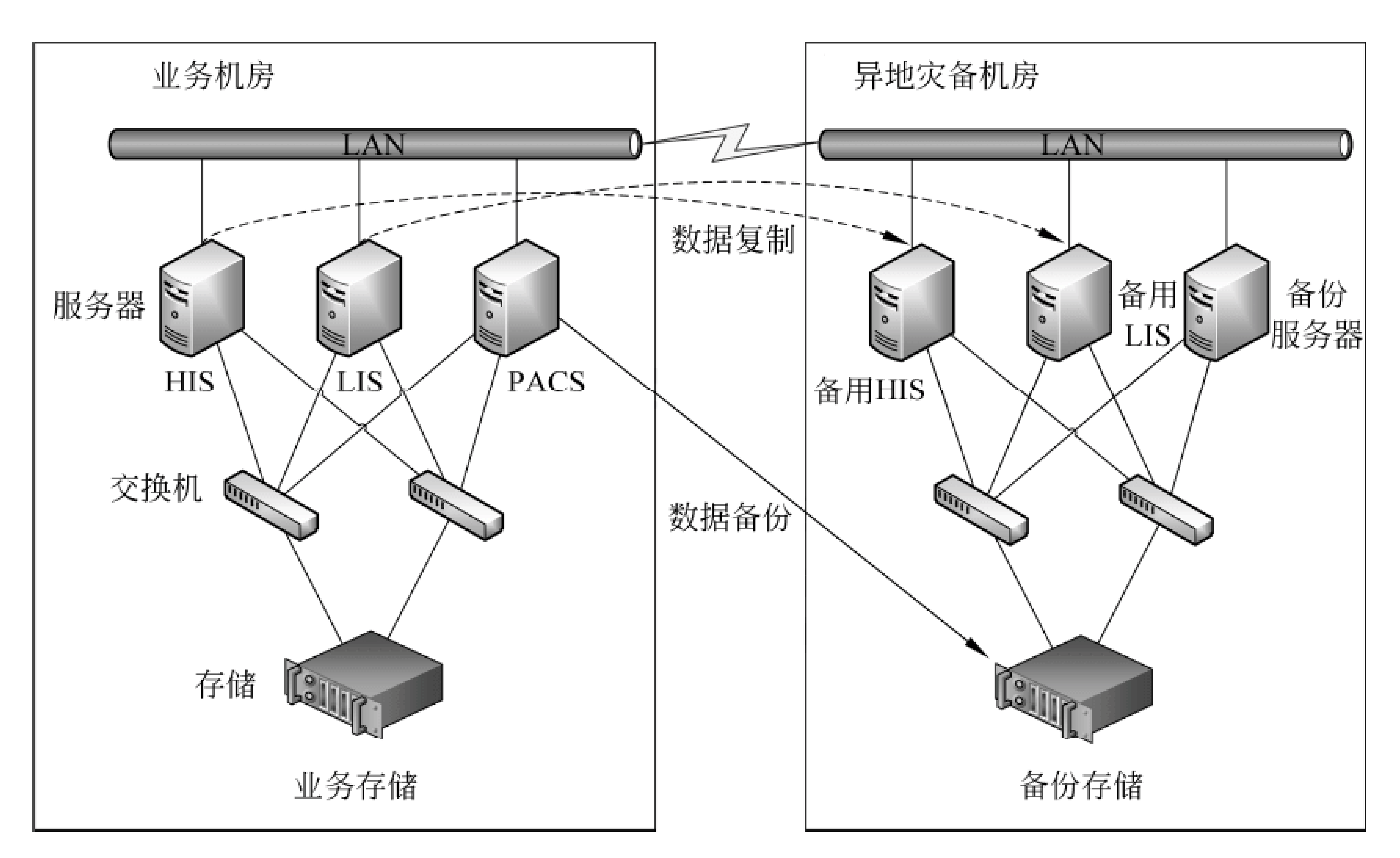


图 8-9　医院的灾备架构

房的备份存储中，提高了其数据安全等级。

该架构的优势如下。

(1) HIS/LIS 在灾备机房有一套完整的应用系统和数据，一旦业务机房发生故障，可迅速将灾备系统激活，让业务流转移到灾备机房，减少业务的停用时间，并且给业务机房足够的维修时间。

(2) 所有系统均通过定时备份的方式把数据保存在异地灾备机房，补足了数据同步无法解决数据逻辑性故障的问题。

(3) 所有数据在异地灾备机房都有一份备份，一旦发生规模型灾难，如火灾、电击、地震等导致业务机房设备大面积损坏，可以确保有一套完整的数据不至于丢失，为灾后重建提供支持。

8.4.3　大型灾备场景

大型灾备场景常见于机构系统非常庞大，系统业务连续性要求和数据安全等级非常高的情况。

以银行为例，银行业在现代经济社会中占据重要地位，每时每刻都在为大众提供服务，一旦银行的系统发生故障而导致业务无法正常运行，所造成的损失和社会影响是无法估量的。

根据银行系统的特点，可以分析如下。

(1) 系统非常庞大，可通过数据双活实现双中心共同运行业务。

(2) 双活中心可解决业务连续性的高要求。

(3) 数据安全等级非常高，可建立异地灾备中心来进行数据的安全保护。

综上所述，银行系统架构可采用两地三中心的方式进行规划，如图 8-10 所示。

图中的 A、B 两中心机房建立在同一个城市的不同位置上，而灾备中心建立在另外一个城市中。A 中心机房和 B 中心机房同时运行不同的业务系统，同时 A、B 两中心机房各自有

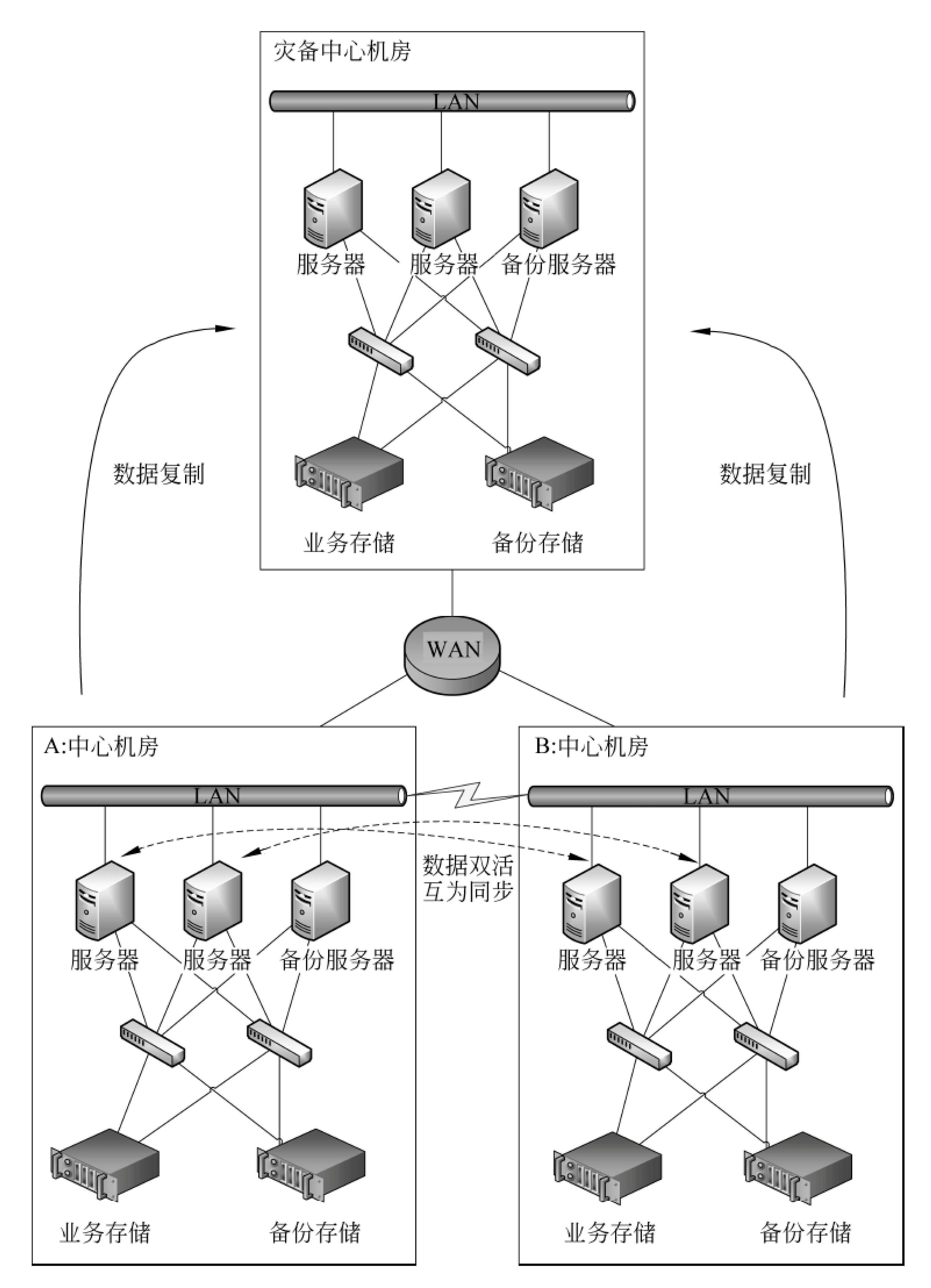


图 8-10　大型灾备场景拓扑图

对方的一套系统，通过数据双活，A、B 两中心机房的数据达到完全一致且都可以实时使用。这种情况下，一旦 A 中心机房发生故障，B 中心机房可马上将 A 中心机房的业务接管过来，保持业务的连续性。

另外，灾备中心有一套完整的 A、B 中心机房的业务系统，如果 A、B 中心机房同时发生故障(如该城市发生强烈地震)，灾备中心可迅速将 A、B 两中心机房的业务接管，防止业务长时间地地断。

该架构的优势如下。

(1) 拥有各种数据保护方式，可以根据不同的灾难级别选择不同的还原操作，使得数据恢复能达到最快和最好的效果。

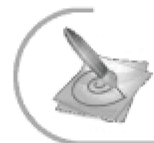
(2) 双活数据中心可以降低用户在业务系统上的投入，提高设备利用效率。

（3）可防止例如地震、泥石流、战争等影响整个城市的大规模灾难，为企业的灾后重建给予强有力的数据保障。

任务拓展

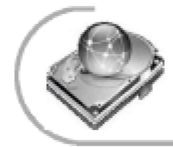
1. 硬件灾备架构和数据灾备架构的区别是什么？
2. 为什么 CDP 无法代替定时备份？
3. 为什么需要进行成本和风险控制？
4. 针对医院的 HIS/LIS 系统应该如何搭建灾备系统？

第9章　数据中心安全运维



任务目标

- 了解数据中心运维体系；
- 了解数据中心运维管理制度；
- 了解数据中心安全管理服务。



项目背景

一套系统从规划诞生到部署实施再到开始使用，只是这套系统漫长生命周期的一瞬，伴随系统终身的是对这套系统的维护。

系统是否能长期稳定地运行，是否能充分满足客户的需求，均离不开系统运维工作，而运维工作中的安全运维则是重中之重。



项目描述

随着聚比特科技有限公司的不断发展，其数据中心业务也越来越多，公司安全运维要求也逐渐提上日程，所以必须制定安全运维办法，来确保数据中心稳定健康地运行。



项目分析

聚比特科技有限公司根据各项国际标准，并吸取国际各大公司的运维管理经验，制定出一套适合自身的安全运维管理办法。



项目实现

根据对国际标准，在充分了解运维管理体系后，制定详细的运维管理制度和安全管理服务。

9.1　运维体系介绍

随着数据中心服务于更多客户，数据中心对灾备中心运营管理有了越来越深的理解。参照ISO27001、ISO9001、ISO20000(ITIL)、ITSS等国际、国内和行业标准以及最佳实践的要求，根据灾难备份和业务连续性的服务重点及运营特点，数据中心建立了更加规范、高效的运营管理体系，用以规范数据中心的日常工作及保持持续稳定的运行。

9.1.1　运维管理服务体系

灾备中心的运营管理服务体系是以国际通行的 IT 服务管理标准——ITIL 为基础，结合灾难备份运营服务的专业特点而建立的。数据中心的运营管理服务体系历经了多个客户、多种平台的灾备系统长期稳定运行的检验，灾备中心的运营管理服务体系日臻完善。

整个服务体系归纳概括为灾备中心运营服务的“三大要求、四大服务方面”，如图 9-1 所示。

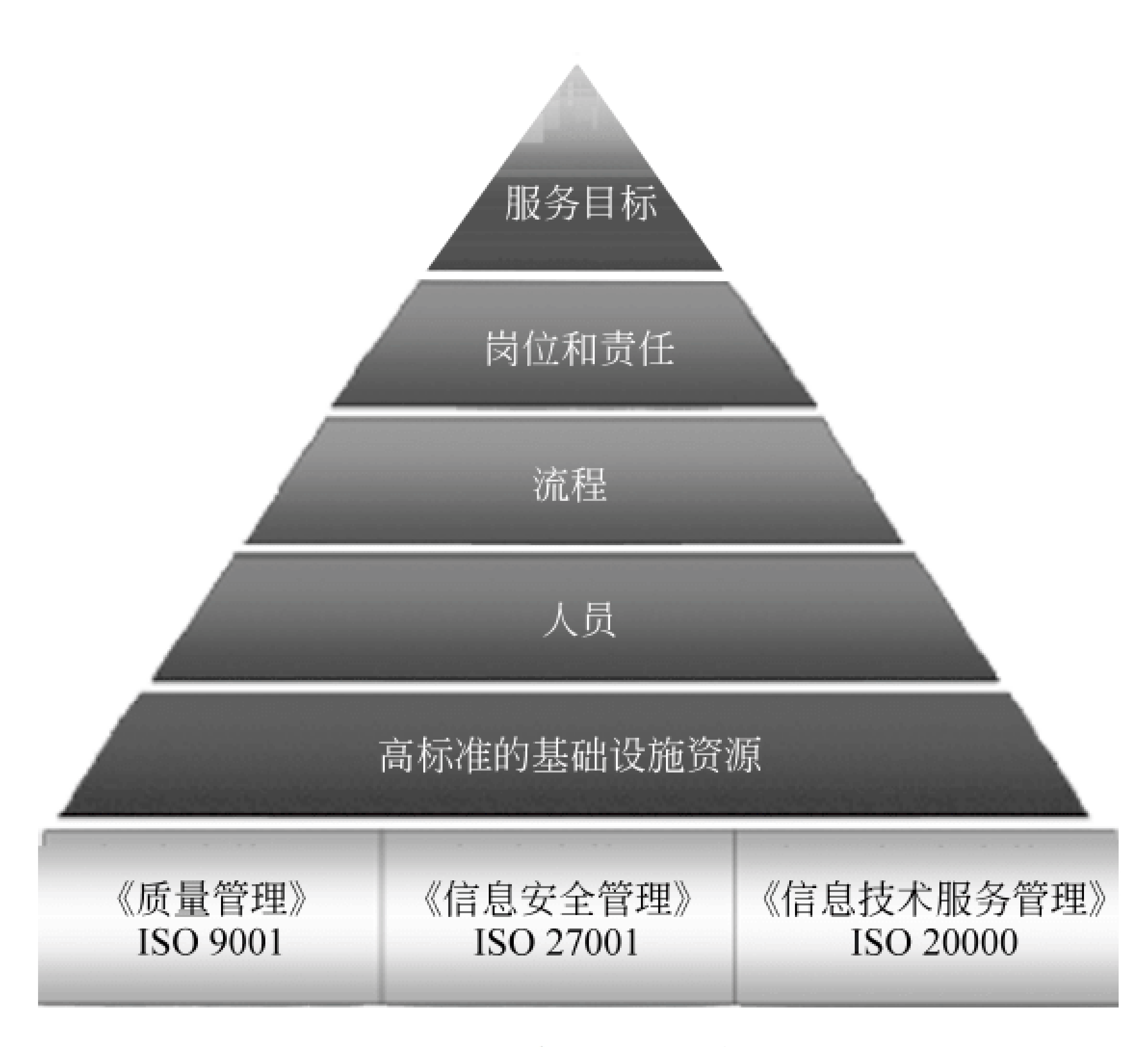


图 9-1　灾备中心运营服务

1. 三大要求

(1) 运营服务的高响应要求——整个服务体系的重中之重，也是对灾难恢复系统运营外包服务公司专业度与服务质量的最直接考验，其关注的是服务的及时性与客户导向。当宣告灾难恢复后，只有高响应度的服务提供商，才能够按既定的服务流程在第一时间为客户提供应急与切换服务，并根据客户的特殊要求提出合理的灾难应急解决方案并接替生产运营，从而最大可能地减轻灾难事件对客户的影响。为此，灾备中心在不同的运营服务阶段提供相应的服务接口与服务人员的组织架构，保证对服务响应度的要求。

(2) 运营服务的高可靠要求——这是整个服务体系的保障，从管理手段和服务流程上保证响应度与可用性的落实。具体体现在对现有人员、资源与技术在执行层面上的标准化、制度化、规范化。只有这样，才能确保当发生不可预测的灾难事件时，灾备系统能够真正起到灾难恢复、业务连续的保障作用。灾备中心的运营服务人员都具备丰富的灾备系统运营管理经验；此外，灾备中心通过采用先进的系统监控技术手段以及严谨的服务流程，有效地保证灾备系统运营服务的可靠性。

(3) 运营服务的高可用要求——这是整个服务体系的基础，从中心资源、业务正常处理流程与人员的支持上，为应急响应、系统切换与接替生产运行的服务工作奠定基础。灾备中心从灾难恢复计划的制订咨询，到灾备系统建设实施的整个过程中，特别是在灾备系统的长期运营服务期间，都充分考虑到对灾备服务可用性的需求，在系统方案、运营服务方案中具

体落实。

2. 四大服务方面

(1) 服务人员——主要指落实包括服务内容、服务接口与服务流程在内的各项具体服务的人员基础。

(2) 服务内容——主要指灾备中心提供的专业灾难恢复系统运营外包服务的具体内容。

(3) 服务流程——主要是灾备中心为规范内部尤其是涉及灾备系统层面所执行的各项工作流程，从制度上保障上述各项服务的顺利提供。

(4) 服务接口——主要指灾备中心提供服务的各种界面，目的是使双方之间的信息与要求能实现无缝衔接。

灾备中心基于 ITIL 标准体系，根据灾备系统运营服务的特点，进一步强化四大服务方面的具体内涵，形成专业灾难恢复系统运营外包的服务体系具体框架，具体如表 9-1 所示。

表 9-1 灾难恢复系统运营外包的服务体系图

服务方面	服 务 细 项
服务内容	日常监控与维护服务、数据验证服务、问题管理、变更管理、安全管理服务、灾难恢复管理服务、灾难恢复预案维护服务、灾难应急及恢复服务、接替生产运行服务
服务人员	日常运营团队、技术支持团队、客户服务团队、应急响应服务团队
服务接口	客户经理、24 小时服务热线、服务报告与会议、应急响应服务
服务流程	事件管理、变更管理、问题管理、应急管理、服务水平管理

3. 运维管理阶段划分

数据中心在服务的具体实施过程中，从服务核心到服务细项形成了完整的服务体系，此外还以时间为主线将灾难恢复系统运营服务提供的过程划分成日常运营、应急与恢复和接替生产运营三个阶段，每个不同阶段的工作重点各不相同，真正切合客户的需求。具体如表 9-2 所示。

表 9-2 不同阶段运营服务

项 目	日常运行服务	灾难应急和恢复服务	接替生产运营服务
服务目标	高可用性，高可靠性	高响应度	高可靠性，高可用性
服务内容	日常监控与维护服务，系统验证服务，安全管理服务，灾难恢复演练服务，灾难恢复预案维护服务	灾难应急及恢复服务	接替生产运营服务
服务人员	日常运营团队，技术支持团队，客户服务团队	应急响应团队，技术支持团队，客户服务团队	运营支持团队，技术支持团队，客户服务团队
服务接口	客户经理，24 小时服务热线，服务报告与会议	24 小时应急服务热线，应急响应服务	客户经理，24 小时服务热线
服务流程	事件管理、变更管理、问题管理，服务水平管理	应急管理	事件管理，问题管理，变更管理

9.1.2　服务提升工具

在信息系统基础运维服务中,涉及对政务资源中心及灾备系统的主机、存储、网络、数据库、中间件等 IT 系统的监控服务,为实现统一监控、集中告警、主动运维的目标,GDS 建议采用业界领先的 IBM Tivoli 产品实现对各种 IT 设备性能监控,采用 BMC Remedy 流程管理平台实现政务资源中心生产和灾备系统的统一运维。

9.1.3　运维监控平台的建设原则

(1) 先进性与成熟性原则

采用先进、成熟和业界领先的 IT 系统监控产品和流程开发工具,建立全面的 IT 资源监控体系和 IT 服务流程管理体系。

(2) 集中统一原则

① 通过统一事件管理平台对各种 IT 资源监控产生的事件信息进行整合并进行统一分析处理。

② 通过综合监视展现平台实现对各种 IT 监控要素的事件信息、性能信息和资产信息进行全方位监控。

(3) 标准性与规范原则

① 指标的规范性:按照监控指标要求实现对各类监控要素的全面监控。

② 接口的规范性:按照规范接口标准实现与生产中心和灾备中心运维监控系统的对接。

③ 流程的规范性:按照 ITIL 的标准和规范建立 IT 服务流程管理体系。

(4) 可定制化原则

① 监控产品的可定制化:除了实现对标准的 IT 资源进行监控外,还应通过定制化手段实现对特殊 IT 对象的监控。

② 监控流程的可定制化:采用先进的流程开发工具开发规范的运维服务流程。

③ 监控界面的可定制化:开发具有政务资源中心特色的统一监控门户,在统一监控界面上采用多角度、多方位的监控视图,全面反映系统的运行状态。

(5) 投资保护与技术延续性原则

政务资源中心的运维监控系统将建立在对现有主机和存储监控的基础上,根据新的服务需求加以扩展和完善,保护原有的投资和技术的延续性。

(6) 自动化与智能化原则

① 事件处理的自动化:对于监控到的重要事件应能自动生产工单,并能自动以多种人性化的方式通知系统维护人员。

② 事件分析的智能化:事件管理模块应具有智能的事件关联分析和事件过滤功能,以方便进行故障定位。

(7) 可扩展与高度集成原则

① 监控系统的可扩展性原则:随着政务资源中心服务职能的扩展,IT 资源将不断扩充,这就要求运维监控系统具有良好的扩展能力,以适应 IT 资源的扩充需求。

② 高度集成原则:IT 运维监控系统将由多个子系统构成,各子系统之间应建立完备的

开发工具和接口规范，以便将各个子系统进行高度集成。

（8）安全性与稳定性原则

① 安全性：由于政务中心的运维监控系统是实现对政务系统及各委办局重要业务系统的监控，一旦系统出现故障，应要求在第一时间予以响应，否则将对业务系统造成严重影响，因此监控系统的安全等级应与 IT 系统的安全要求相适应。

② 稳定性：未来重要的政务系统将需要 7×24 小时连续运行，任何时刻 IT 系统出现故障时，运维监控系统应立即通知维护人员予以解决，为保证系统监控的有效性和响应的及时性，运维监控系统应具有良好的容错能力，保证运维监控系统运行的稳定性。

③ 生产系统运行的稳定性：IT 系统监控需要在监控对象中安装监控代理，以实时地收集性能数据，对生产系统的性能造成一定的影响，因此运维监控系统应保证对生产系统资源占用最小，确保对生产系统正常运行影响最小。

9.1.4 系统集中监控方案

1. 主机系统监控

目前 GDS 的数据中心采用 IBM Tivoli Monitoring 来实现对主机可靠性管理，针对政务资源中心的主机系统，GDS 将在政务资源中心的主机上安装 ITM 监控代理，收集主机的性能、状态和警告信息，并发送到后台的监控服务器上，实现对所有主机的统一监控。当服务器系统发生问题时，可以迅速报警，使管理员可以及时获知系统出现的问题，通过 IBM Tivoli Monitoring 对系统重要的资源进行监控，并定义门限值，一旦发现超过门限值，则将事件通知管理员。为保证系统的可靠性，会对服务器的一些重要资源进行监控，包括：①服务器的状态；②重要进程状态；③磁盘空间；④性能超过指标；⑤文件的修改情况；⑥日志文件的变化情况。

管理员定义对这些参数的门限值，可以根据不同的值定义不同的警告级别和相应的报警方式，如磁盘空间利用率达到 80%为 Warning（警告）状态，记录日志文件；90%为 Critical 状态，报警到故障控制台。

2. 存储系统监控

在政务资源数据中心的系统中，大部分服务器与存储设备均通过光纤交换机以 SAN 的方式进行连接，即在服务器与存储设备之间构成了一个存储区域网络。由于 SAN 网络的使用，必须建立相应的管理维护机制和工作管理流程，才能充分发挥 SAN 网络和数据集中的优势。通过 SAN 上异构平台的数据共享，提高存储效率。通过存储域网络管理软件，监控存储域网络的状态，及时预报和发现故障，监控网络性能。通过存储域网络管理软件直接设置和配置存储设备的连接和分配，通过软件完成一些以前需手工维护的工作。同时，通过存储域网络资源管理软件，分析解决存储域网络可能出现的问题，生成分析报表，使管理人员和维护人员对存储系统的状况了如指掌。

SAN 存储系统管理主要由四部分组成，具体描述如下。

（1）存储网络管理

① 发现代理程序。通过内部管理和外部管理两种方式实现对 SAN 网络的信息搜索。内部管理是指搜集流过 SAN 本身的信息；外部管理是指通过 TCP/IP 连接，通常通过

SNMP MIB 来搜集信息。

② SAN 网络拓扑管理。SAN 网络拓扑管理将以图形的方式显示了 SAN 结构中的所有部件，如磁盘阵列、光纤磁带库、主机系统、HBA 卡、光纤交换机、光纤网关、光纤连接等。

(2) 存储设备管理

① 性能管理。SAN 存储系统的性能管理主要包括磁盘存储服务器的性能和 SAN 网络性能。存储网络性能监控的主要内容包括：

- I/O 请求的数量；
- 确认每天最忙的时间；
- 传输的数据量；
- 物理 I/O 的读写响应时间；
- 确认最忙的阵列、适配卡和服务器；
- Cache 使用的统计数据；
- 在现有主机工具提供增加的信息。

② 磁盘访问权限控制。在 SAN 存储网络中，对于不同的磁盘系统的访问控制，需要相应的硬件分配软件，包括：

- 每一个服务器的存储容量(已用的和未用的)；
- 存储服务器的存储随时间增长的图表；
- 总结主机及其在存储服务器上的存储；
- 多台主机共享的容量细节。

(3) 存储资源管理

① 开放性、标准化及弹性化的架构；

② 单一界面的跨平台分析；

③ 存储容量规划；

④ 档案层次分析；

⑤ 自动化事件关联；

⑥ 符合客户需求的报表；

⑦ 资料库分析；

⑧ 可用度报告。

(4) 存储系统故障管理

存储系统的故障总是与整个 IT 系统的其他部分互相关联、相辅相成的。所以我们不能只从存储的单一系统来处理问题，需要将存储管理和系统管理统一起来处理问题、分析问题，评估服务水平、分析故障对于业务的影响。存储系统管理通过统一的事件处理平台，实现对故障的统一分析处理、事件关联和根源分析。

3. 网络系统监控

通过 IBM Tivoli NetView 网管软件的轮询或请求/应答方式对政务资源中心的网络设备进行监控，网上 IP 资源会在需要的时候发出相应的报告，IBM Tivoli NetView 可以持续不间断地对网上 IP 资源的状态、配置和事件进行监控，甚至连一个参数的改变 NetView 都可以收到报告。

通过 SNMP 的 AGENT，网络管理人员可以监控到许多网络设备性能的参数，如网络路由器的数据传输量、坏包的数量等影响网络效率和质量的参数，从而为诊断系统故障提供有力的工具。

在 IBM Tivoli NetView 中可以对某些监控的对象，如路由器端口的网络流量等，设置阈值。当一旦达到所设定的阈值时，NetView 会报警、自动执行相应的处理等。如可以监控网络通信线上的数据通信量及重要的文件服务器上的硬盘空间利用率，当利用率达 80％时可自动向网管中心发出报警信息，使系统管理员及时采取措施避免网络故障，减少网络重大故障的发生概率。

4. 数据库系统监控

通过 ITM for Database 可以实现对 Informix、DB2、SQL Server、Oracle 等多种数据库的监控。

(1) DB2 数据库的监控

ITM for DB(Oracle)提供了以下预先定义的监控项目，按照属性相关性分为 Application、Buffer Pool Data、Database、Locking Conflict、System Overview 和 Tablespace 共 6 个属性组，每个属性组包含一个或多个预先定义的属性(即监控项目)，用户可以直接使用，也可以更改或者新建。

(2) 对 MS SQL Server 的监控

MS-SQL 提供了以下预先定义的监控项目：Database 信息、Device 信息、进程信息、锁信息、状态信息、Server 信息等。

(3) 对 Oracle 的监控

Oracle 提供了诸如高级消息队列、告警日志详细及汇总、缓存使用信息监控、配置信息、锁争夺信息、数据库、文件等预先定义的监控信息。

5. 中间件系统监控

对于中间件系统监控，可通过 IBM 复合应用管理系统 IBM Tivoli Composite Application Management(ITCAM)来实现，ITCAM 应用监控管理软件对应用系统实时监控，分析运行效率，提供灾难预警，准确定位错误，提供有效维护的建议，在很少占用系统资源模式下可以长期以 7×24 方式运行，可以产出包含多种图、表和文字等实时性、历史周期性、前瞻预期性的报告，并支持打印和归档。ITCAM 应用监控管理软件本身应易于安装、维护和使用，并且功能强大。能够通过较低的系统资源占用收集数据库运行状态的实时数据，其采样频率可以由系统动态调节，也可以由用户自定义，适宜 7×24 的方式运行。收集的信息要全面和丰富，不仅有实时信息，也能方便地存储和分析历史信息。

通过 ITCAM 产品能够监控 MQ、TongWeb/Q、CISC、WebSphere、Tomcat、WebLogic、IIS 的运行状况和日志文件的变化情况；提供应用服务器日志监控，包括 Message ID、Message 内容、进程号、严重级别、发生时间、Job 名称和 ASID 等信息；提供 Log Analysis 分析试图，提供详尽的日志分析报表，系统地监控应用服务器和应用程序的 Log；同时提供对基于 WebSphere、Weblogic、Tomcat 的应用服务器进行实时监控和历史数据分析，它能够发现并且报告 J2EE 应用的效果，获取中间件的重要性能指标。它的监控贯穿整个应用流程，如应用程序服务器、中间件适配器、传输协议、数据库等。对于 WebSphere、Weblogic、Tomcat 中间件的性能监控，提供系统级别的数据，例如，应用服务器的状态、中央处理器的

使用、内存的使用、数据库连接池、JVM 线程池、EJB 的使用等，也会被收集，用来辅助用户去分析问题、解决问题。

9.1.5　统一事件管理平台建设

针对政务中心的资源情况，采用 IBM Tivoli Enterprise Console(TEC)统一事件管理平台实现统一事件管理。

1. 事件采集

TEC 能够支持多种的事件采集方式，提供多种的事件适配器能够支持对 SNMP 事件、主机系统日志事件(syslog)、第三方可集成产品(如 HP NNM、Vertias)管理事件的集成。TEC 提供了事件发送方式，可以非常容易地将其应用事件、非标准事件传送到 TEC 中。TEC 提供了多种抗事件风暴的能力，通过内存缓存、硬盘缓存来建立事件缓存队列，从而防止管理事件的丢失。

2. 事件分析

TEC 独有的基于规则的分析引擎具有多重的事件关联分析能力，能够按照时间、事件属性、事件类别、来源进行跨资源的分析，帮助进行问题事件的查找。TEC 新的 ITS (Integrated TCP/IP Service)部件是一套将网络和主机事件进行集中关联分析的规则，加载了 ITS 规则库后，TEC 能够自动分辨是网络还是主机故障造成无法提供有效的服务，从而缩小了故障定位的范围。

TEC 特有的分布处理部件(AIM)，能够对区域事件进行规则处理，而所有的规则都来自于中心的 TEC 系统。通过 AIM 部件的分布处理，可以形成多级事件处理机制，得以减轻中心事件服务器的压力，从而可以支持更大的规模和更为复杂的关联规则。

3. 事件处理

TEC 的事件处理机制可以根据多种条件触发不同的处理方法，包括告警方式、自动脚本调用方式等。运行管理人员可以在 TEC 的事件管理窗口进行事件的交互式处理，从而可以通过手工方式对特定事件进行处理。TEC 支持将事件送到故障管理流程平台中，包括 BMC Remedy、基于 Domino 的故障管理系统等。

4. 事件查看

运行管理人员可以分成不同的组，每个组可以查看、处理不同类型的事件，从而进行分区域管理。运行管理人员可以使用 TEC 的 Java Console 实现从远程来访问 TEC 服务器，从而实现方便的管理。

9.1.6　报表管理系统

报表系统对于大数据量的处理，数据的准确性等方面均有优异性能，后台的数据库一个地点一套，保障数据的可靠性与及时性，为不同层面的人员设计不同的报表，并推荐报表上报制度，对领导决策提供依据；对于系统故障、性能数据都会存储在后台的数据库中，通过报表系统的统计分析处理。管理员可对故障、性能管理数据库中存储的历史记录进行访问，生成分析报告。

9.1.7 运维服务管理平台建设

GDS 的数据中心统一采用 BMC Remedy 流程引擎作为数据中心运维流程管理平台，为提高政务中心的服务质量和服务水平，GDS 将 Remedy 平台延伸到政务中心，不仅可解决 GDS 工程师的运维问题，而且还可汇集 GDS 其他数据中心和后台专家共同解决问题，实现政务资源中心的多级运维模式。

BMC Remedy IT Service Management 是第一个经 Pink Elephant 的 Pink Verify 和 OGC 认证程序认证且符合 ITIL 兼容性功能要求的整体解决方案。

该方案提供了一套安装即用的集成功能，包括由 ITIL 所指定的服务台功能、事故管理功能、问题管理功能、服务请求管理、变更管理以及配置管理等功能。

基于 Remedy 的解决方案提供了可以应用到每个应用程序中且与 ITIL 兼容的最佳实践。并且这些应用程序可以根据独特的服务支持流程和工作流进行轻松修改，以便更好地满足 ITSM 功能需求。其产品架构如图 9-2 所示。

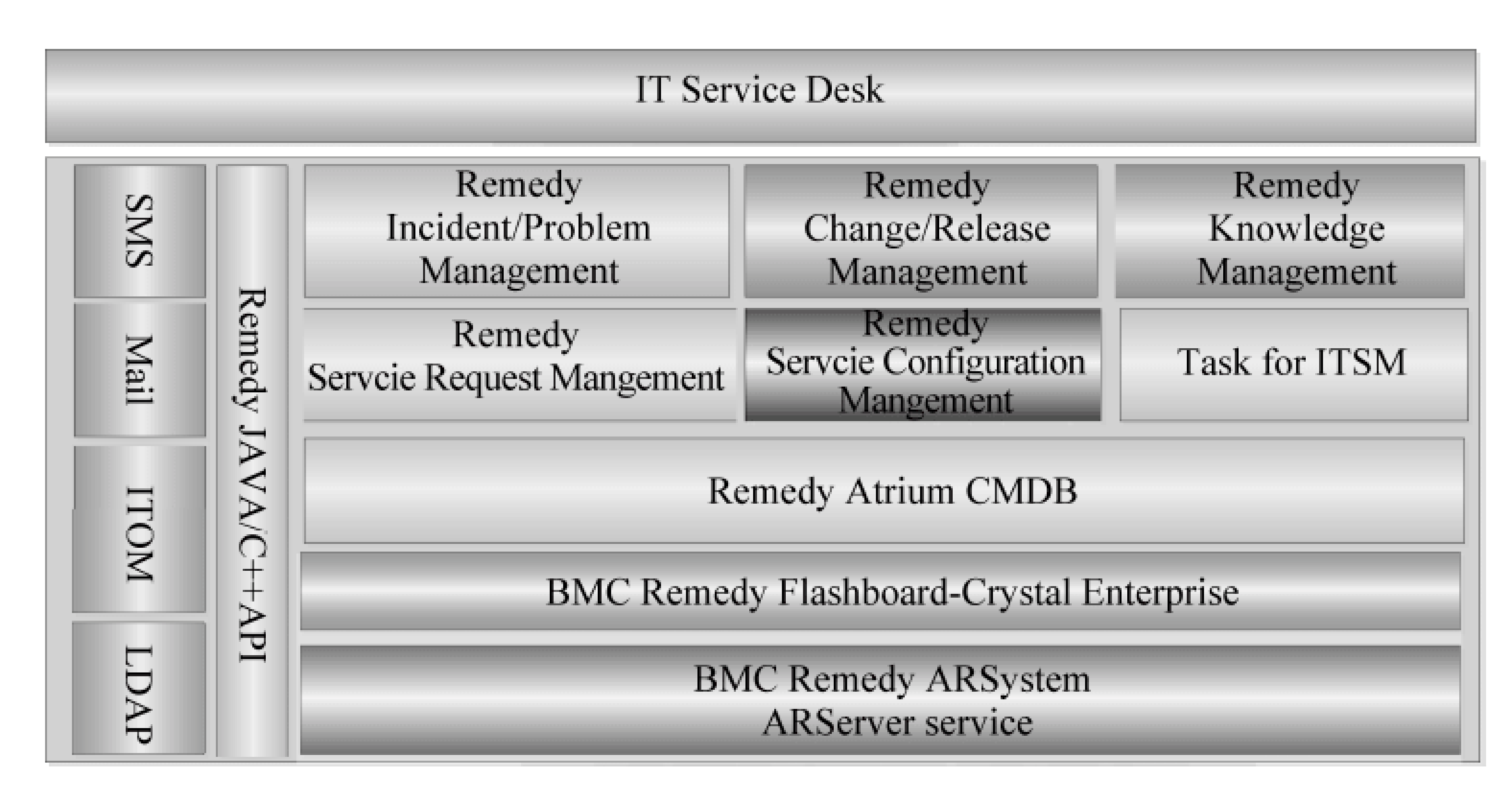


图 9-2 BMC ITSM 产品架构

本项目应用管理流程包括：服务台、事件管理、问题管理、配置管理、知识库管理、服务请求管理及日常任务等，这些流程在 BMC Remedy 产品架构中都有所体现，如图 9-2 所示的这些功能模块都采用了 ITIL 最佳实践，并基于 BMC Remedy 强大的工作流引擎平台 ARS 构建而成。在业界的评测中 BMC 是第一个通过全部 ITIL13 个流程的产品。在 BMC ARS ITIL 工作流引擎平台中实现个性化的 IT 服务管理流程。针对政务资源中心的运维管理流程，选用以服务支持核心模块为主、其他支持流程为辅的流程框架，如表 9-3 所示。

表 9-3 流程框架图

本次项目流程	BMC Remedy 产品实现
事件管理(服务台)、问题管理	BMC Remedy Service Desk
自服务与服务请求	BMC Service Request Management
变更管理、发布管理	BMC Remedy Change Management

续表

本次项目流程	BMC Remedy 产品实现
知识库管理	BMC Remedy Knowledge Management
配置管理	BMC Remedy Asset Configuration Management
日常任务管理	通过 BMC Remedy AR System 进行定制开发来实现此功能
基础平台	BMC Remedy AR System

BMC Remedy IT Service Management 的服务台与事件管理流程的主要功能满足 ITIL 的标准，能够实现尽快解决影响应用系统正常运行的事件，保持业务支撑系统的稳定性。问题管理流程的主要功能也满足 ITIL 的标准，能够实现问题管理流程的根本目标。即根本目标是消除或减少生产环境中事件发生的数量和严重程度，从而为企业建立一个稳定的 IT 环境，提高 IT 服务的可用性。

BMC Remedy IT Service Management 的自服务与服务请求功能模块满足 ITIL 的标准，能够实现用户在很少甚至无须服务台干预的情况下请求服务和查找信息，从而减轻运维人员的工作压力。

BMC Remedy IT Service Management 的变更/发布管理流程功能模块满足 ITIL 的标准，通过一系列的控制措施和流程，确保对生产环境风险的控制，提高企业资源使用率。

BMC Remedy IT Service Management 的配置管理主要功能满足 ITIL 的标准，能够实现配置管理通过对各配置项的定义、管理以及统计分析来提高 IT 环境的可视化，降低 IT 成本、增加投资回报，并确保 IT 环境的稳定性。

BMC Remedy IT Service Management 的知识库管理流程主要功能满足 ITIL 的标准，能够实现知识库的管理需求。可以通过知识库管理在故障自动处理和人工处理的过程中，在知识库中得到相关故障维护的分类和快速定位，找到匹配的处理案例，便于处理人进行借鉴，进行知识的总结与归类并应用知识。

BMC Remedy AR System 平台具有很好的稳定性、开放性和广泛的集成性，易于进行二次开发。在 ARS 平台上实现作业调度以及日常检查功能，有利于规范和简化光大银行的日常运维工作，提高工作效率。

9.1.8　应急机构与职责

1. 应急领导小组

应急领导小组是市局信息系统应急管理工作的领导机构。主要职责如下。

(1) 审批信息管理部门的申请。

(2) 在网络与信息系统安全事件发生时，组建应急指挥部，指挥、协调信息系统安全事件 应急工作。

(3) 指导基层单位网络与信息系统安全事件的应急工作。

(4) 宣布进入和解除应急状态，决定实施和终止网络与信息系统安全事件应急预案。

(5) 统一领导Ⅰ级和Ⅱ级网络与信息系统安全事件的应急处置工作。

(6) 研究决定对外有关网络与信息系统安全事件的新闻发布。

2. 信息管理部门

信息管理部门是网络与信息系统安全事件的保障管理部门。主要职责如下。

(1) 负责应急物资储备、预案演练、培训计划申请。

(2) 落实应急领导小组部署的各项任务，并向应急领导小组报告应急处置中发现的问题。

(3) 监督执行应急领导小组下达的应急指令、重大应急决策和部署，协调各方应急资源，组织应急处置。

(4) 按相关规定参与、配合网络与信息系统安全事件调查，总结应急处理经验和教训。

3. 应急工作组

信息管理部门根据事件类型和应急处置的实际需要，组织内部和外部人员建立网络与信息系统安全应急事件故障处置及技术支持工作组。具体职责为：

(1) 执行信息管理部门下达的应急处置工作和保障任务；

(2) 执行应急处理、控制事件范围、进行事件恢复；

(3) 提供技术支援、协助事件调查。

4. 数据恢复组

数据恢复组由成都政务数据资源中心技术人员组成。数据恢复组的主要负责如下。

(1) 突发事件发生时

① 负责接受和确认灾备中心数据恢复准备和数据恢复请求通知，并按通知的要求或预先设定的检查程序完成数据存储与备份系统状态检查等数据恢复的准备工作。

② 负责实施灾备端备份数据恢复至生产端指定区域。

③ 在数据恢复过程中协助应急工作组其他团队完成恢复工作。

(2) 日常相关工作

① 负责日常基础运营服务。

② 负责日常存储资源管理，对存储灾备系统的存储资源进行监控、管理。

③ 负责根据成都市工商局的需求，完成存储资源的分配与回收。

④ 负责数据一致性验证和可用性验证。

⑤ 负责《数据恢复预案》的维护和更新。

9.1.9 突发事件分级

1. 事件分级说明

根据网络与信息安全事件对服务的社会用户和内部用户的影响范围、程度、可能产生的后果和损失等因素，将信息系统事件分为Ⅲ级、Ⅱ级和Ⅰ级三个等级。

发生Ⅲ级事件进入信息系统预警状态，发生Ⅱ级信息系统事件进入信息系统Ⅱ级应急状态，发生Ⅰ级信息系统事件进入Ⅰ级应急状态。

各基层单位应参照市局网络与信息系统安全事件分级标准，结合本单位的实际情况，制定相应预警状态和应急状态标准。但各单位分级标准不能高于市局同级标准。

2. Ⅲ级信息系统事件

(1) 对市局及所属各单位，因下列原因对所服务社会用户的生产、生活造成影响，影响用户数量超过本单位服务总用户数量的20%，低于50%。

① 通道与网络故障。
② 主机设备、操作系统、中间件和数据库软件故障。
③ 应用停止服务故障。
④ 应用系统数据丢失。
⑤ 机房电源、空调等环境故障。
⑥ 大面积病毒爆发、蠕虫、木马程序、有害移动代码等。
⑦ 非法入侵,或有组织的攻击。
⑧ 自然灾害或人为外力破坏。
⑨ 其他原因。

(2) 对市局及所属各单位,因下列原因对本单位的生产、经营、管理和信息发布造成影响,影响内部用户数超过 20%,低于 50%。

① 通道与网络故障。
② 主机设备、操作系统、中间件和数据库软件故障。
③ 应用停止服务故障。
④ 应用系统数据丢失。
⑤ 机房电源、空调等环境故障。
⑥ 大面积病毒爆发、蠕虫、木马程序、有害移动代码等。
⑦ 非法入侵,或有组织的攻击。
⑧ 自然灾害或人为外力破坏。
⑨ 信息发布和服务网站遭受攻击和破坏。
⑩ 其他原因。

(3) 市局及所属各单位出现大面积的有害信息传播,影响范围大,影响各单位内用户数超过 30%,低于 50%。

3. Ⅱ级信息系统事件

(1) 对市局及所属各单位,因下列原因对所服务社会用户的生产、生活造成严重影响,影响用户数量超过本单位服务总用户数量的 50%,低于 90%。

① 通道与网络故障。
② 主机设备、操作系统、中间件和数据库软件故障。
③ 应用停止服务故障。
④ 应用系统数据丢失。
⑤ 机房电源、空调等环境故障。
⑥ 大面积病毒爆发、蠕虫、木马程序、有害移动代码等。
⑦ 非法入侵,或有组织的攻击。
⑧ 自然灾害或人为外力破坏。
⑨ 其他原因。

(2) 对市局及所属各单位,因下列原因对本单位的生产、经营、管理和信息发布造成严重影响,影响内部用户数超过 50%,低于 90%。

① 通道与网络故障。
② 主机设备、操作系统、中间件和数据库软件故障。

③ 应用停止服务故障。

④ 应用系统数据丢失。

⑤ 机房电源、空调等环境故障。

⑥ 大面积病毒爆发、蠕虫、木马程序、有害移动代码等。

⑦ 非法入侵,或有组织的攻击。

⑧ 自然灾害或人为外力破坏。

⑨ 信息发布和服务网站遭受攻击和破坏。

⑩ 其他原因。

(3) 市局及所属各单位出现大面积的有害信息传播,影响范围大,性质恶劣,影响各单位内部用户数超过 50%,低于 90%。

(4) 涉及国家或单位利益的秘密信息通过信息系统泄露,造成重大影响。

4. Ⅰ级信息系统事件

(1) 对市局及所属各单位,因下列原因对所服务社会用户的生产、生活造成特别严重影响,影响社会用户数量超过本单位服务总用户数量的 90%。

① 通道与网络故障。

② 主机设备、操作系统、中间件和数据库软件故障。

③ 应用停止服务故障。

④ 应用系统数据丢失。

⑤ 机房电源、空调等环境故障。

⑥ 大面积病毒爆发、蠕虫、木马程序、有害移动代码等。

⑦ 非法入侵,或有组织的攻击。

⑧ 自然灾害或人为外力破坏。

⑨ 其他原因。

(2) 对市局及所属各单位,因下列原因对本单位的生产、经营、管理和信息发布造成特别严重的影响,影响内部用户数超过 90%。

① 通道与网络故障。

② 主机设备、操作系统、中间件和数据库软件故障。

③ 应用停止服务故障。

④ 应用系统数据丢失。

⑤ 机房电源、空调等环境故障。

⑥ 大面积病毒爆发、蠕虫、木马程序、有害移动代码等。

⑦ 非法入侵,或有组织的攻击。

⑧ 自然灾害或人为外力破坏。

⑨ 信息发布和服务网站遭受攻击和破坏。

⑩ 其他原因。

(3) 市局及所属各单位出现大面积的有害信息传播,影响范围大,性质恶劣,影响本单位内部用户数超过 90%。

(4) 涉及国家或单位利益的机密信息泄露,造成特别重大影响。

5. 突发事件升降级说明

突发事件的处理是一个发展变化的过程，每隔一段时间对事件的影响程度和范围进行重新评估，按照上述事件分级的定义重新判定事件级别。

突发事件的升级策略如图 9-3 所示。

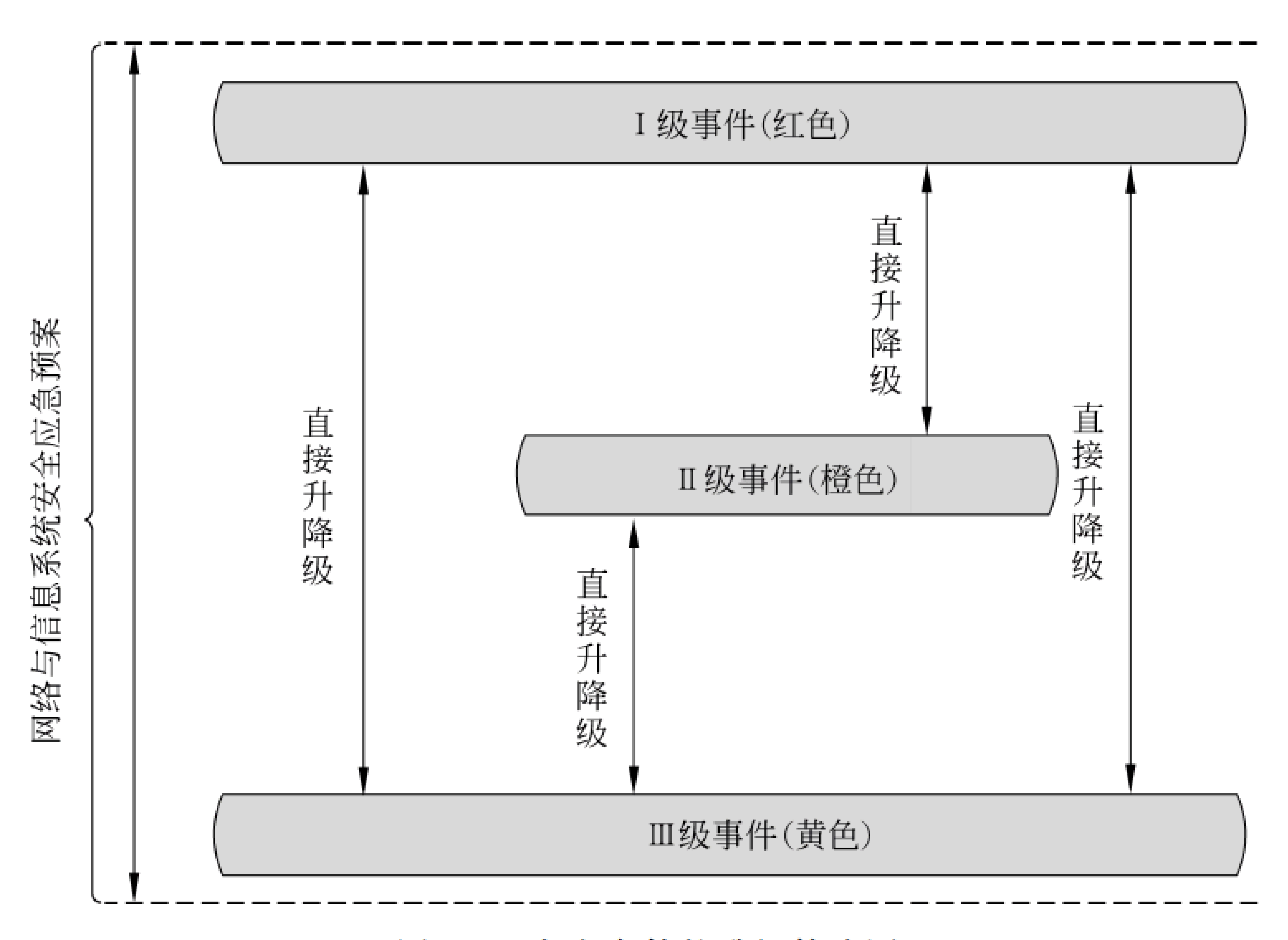


图 9-3　突发事件的升级策略图

9.1.10　应急响应

1. 应急启动

发生网络与信息系统安全事件后，事件发生单位应立即启动应急预案，本着尽量减少损失的原则，将应急事件尽快隔离，在不影响正常生产、经营、管理秩序的情况下，保护现场。

信息管理部门接到各单位网络与信息系统安全事件的应急报告后，根据事件情况，启动信息系统安全应急预案。

信息管理部门接到Ⅰ级和Ⅱ级事件报告后，根据事件的性质和影响向应急领导小组报告。

2. 事件报告

发生网络与信息系统安全事件时，由各级信息管理部门逐级报告。

报告分为紧急报告和详细汇报。紧急报告是指事件发生后，各级信息管理部门向上级信息管理部门以口头和应急报告表的形式汇报事件的简要情况；详细汇报是指由相应单位信息系统应急处理机构在事件处理暂告一段落后，以书面形式提交的详细报告。

各单位信息管理部门对各类事件的影响进行初步判断，有可能是Ⅰ级事件的，须在 30 分钟内向市局信息管理部门进行紧急报告，Ⅱ级事件应在 60 分钟内进行报告，Ⅲ级事件在 3 小时内汇报。

发生下列情况引起管理信息系统事件时，各单位须向市局信息管理部门报告。

(1) 大面积病毒爆发，且快速扩散事件。

(2) 对主要网站、应用系统和关键设备等的大规模攻击和非法入侵,攻击数据包源 IP 地址不明或为内部 IP 地址。

(3) 有害信息通过电子邮件等方式在内部网络上大面积传播。

(4) 信息网络上传播不符合国家和单位保密要求的国家和单位涉密信息事件。

(5) 其他从一个单位发生且能够影响其他单位和整个成都市工商信息系统事件。

任何单位和个人均不得缓报、瞒报、谎报或者授意他人缓报、瞒报、谎报事件。

事件报告的内容和格式要求如下。

(1) 各单位要规范口头报告的内容和格式,要求内容简洁、清楚、准确。

(2) 口头报告的内容主要包括事件发生的时间、概况、可能造成的影响等情况。

(3) 口头报告后应用传真方式报送市局信息管理部门。

3. 应急处置

网络与信息系统安全事件应急处置应按照各专业协同处理的原则进行,需要内部多个部门和专业协同处置或外部应急资源支持的应急事件,由信息管理部门负责统一协调。

信息管理部门以保障重要应用系统和信息网络及基础应用的安全稳定运行为目标。当发生病毒、非法入侵、网络攻击、有害信息传播、不符合规定的涉密信息传播等事件时,迅速调整网络安全设备的安全策略或隔离事件区域,查找源头,采取有效措施,控制事件的发展。当管理信息系统出现软硬件设备故障、网络链路故障、机房环境设备故障等事件时,应立即启用备份系统和备用设备,调整系统运行和安全策略,恢复系统正常运行。

发生Ⅲ级信息系统事件后,事件相关单位应立即启动相关应急预案和专项应急预案,根据事件原因采取相应措施控制影响范围,同时向市局信息管理部门报告,市局信息管理部门通知相关单位、部门和工作组启动应急准备工作。

信息系统事件由Ⅲ级发展为Ⅱ级或发生Ⅱ级事件后,事件相关单位应立即启动相关应急预案和专项应急预案,开展应急处理。市局信息管理部门接到应急报告后,根据事件产生的原因协调相关资源,支持事件相关单位应及时、有效地进行处理,控制事件发展,同时上报市局应急领导小组,市局应急领导小组协调其他应急资源支持应急处理。

事件由Ⅱ级发展为Ⅰ级或发生Ⅰ级事件后,事件相关单位应立即启动相关应急预案和专项应急预案。市局信息管理部门接到应急报告后,根据事件产生的原因协调相关资源,组织有关各方对事件进行及时、有效的处理,控制事态发展,同时上报市局应急领导小组。

当出现自然灾害、恐怖袭击、战争、人为非法破坏等重大事件时,以及发生大规模的计算机病毒爆发、网络攻击、内部人员重大作案等重大网络与信息系统安全事件时,由于重大技术故障导致信息网络与重要信息系统无法正常运行事件,无法迅速恢复正常生产、经营和管理工作时,由市局应急领导小组上报上级安全应急办公室,请求上级单位、公安部和信息产业部的应急支持。

4. 应急结束

在同时满足下列条件下,事件相关单位应急领导小组或信息管理部门可决定宣布解除应急状态。

(1) 各种网络与信息系统安全事件已得到有效控制,情况趋缓。

(2) 网络与信息系统安全事件处理已经结束,设备、系统已经恢复运行。

（3）上级应急部门发布的解除应急响应状态的指令。

事件相关单位应急部门应及时向现场应急工作组和参与应急支援的有关单位传达解除应急状态响应的指令，恢复正常生产工作秩序。

5. 后期处置

（1）后期观察

Ⅰ级网络与信息系统安全事件应急处理结束后应密切关注、监测系统 2 周，确认无异常现象。

Ⅱ级网络与信息系统安全事件应急处理结束后应密切关注、监测系统 1 周，确认无异常现象。

Ⅲ级网络与信息系统安全事件应急处理结束后应密切关注、监测系统 2 天，确认无异常现象。

（2）调查评估

网络与信息系统安全事件应急处理结束后，影响到公众利益和国家安全的事件，按照国家相关部门的要求配合进行事件调查。

对按照市局相关规定要求需要市局成立调查组的事件，由市局组织成立调查组，对事件产生的原因、影响进行调查和评估，对责任进行认定，提出整改建议。

按照市局相关规定由各单位自行组织调查的，各单位对事件产生的原因进行调查，对产生的影响进行评估，对责任进行认定，提出整改措施。调查报告按市局规定上报有关部门，同时报送市局信息管理部门。

（3）改进措施

网络与信息系统安全事件处理结束后，相关单位应组织研究事件发生的原因和特点、分析事件发展过程，总结应急处理过程中的经验和教训，进行应急处置知识积累，进一步补充、完善和修订相关应急预案。相关单位应结合运行过程中的异常和相关事件，综合分析信息系统中存在的关键点和薄弱点，提出该类事件的整改措施，制订整改实施方案并予以落实，整改措施和方案报信息管理部门备案。

9.2 运维制度管理

9.2.1 管理制度架构

数据中心已经依据 ISO27001、ISO9001、ISO20000 等国际和行业标准以及最佳实践的要求，建立了一套制度化、流程化、标准化的数据中心运营管理体系，用以规范数据中心的日常工作，并持续稳定运行，该管理体系通过了专业认证并在长期的实施运行中不断的持续完善，其框架如图 9-4 和表 9-4 所示。

9.2.2 管理制度说明

灾难备份系统日常运营管理的好坏，对项目的成功与否起着至关重要的影响。灾难备份系统中的设备平时处于备援状态，当灾难发生时，为保证其能接替总行数据中心的运行，需要灾难备份系统具有非常高的可用性和可靠性；不仅如此，当数据中心面向灾难备份系统

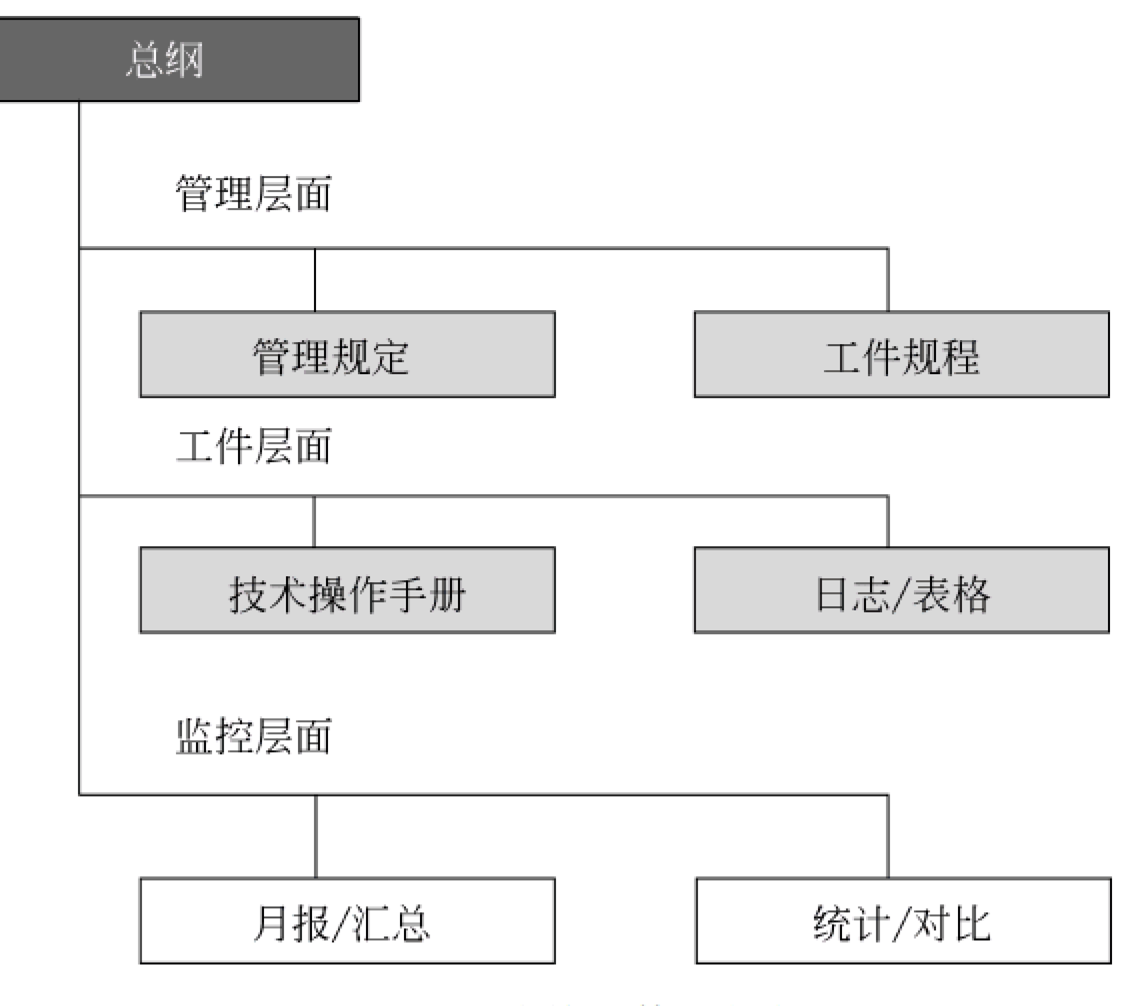


图 9-4　运营管理体系框架

表 9-4　运营管理体系框架说明

分　　类	内　容　说　明
管理制度总纲	这是一个纲领性文件，主要明确和描述备份中心的职责、工作目标、主要原则和工作内容，并对岗位设置和岗位职责和主要的管理原则进行界定
岗位职责描述	针对生产中心的工作特点，对所需完成的各类工作确定岗位，并对各岗位的职责、工作内容、工作规范和管理制度进行明确和描述
工作规程和管理规定	对整个生产中心范围的工作内容和有关的管理规定和工作规范和流程进行说明和明确，如安全管理规定、系统变更管理规定、生产故障处理和管理规定、机房进入管理规定等各方面的管理规定和制度
操作手册	对具体的工作过程和操作命令序列，如能建立技术说明和操作手册均应尽可能建立，特别是日常经常需要进行的操作和在紧急状态下的操作过程均应建立操作手册，以确保有关操作和过程正确无误、稳定可靠
日志记录	对日常巡检监控、设备系统的操作维护、人员设备的进出等建立日志记录表格，进行提醒、记录和检查，每天的日志记录表格需要操作人员、操作领班、值班经理和有关人员进行记录、检查和复核，确保每天的工作有序地进行和可追踪
月报/统计报表	对各项工作需要有一定形式的月报等统计表格，如每月基础环境运行、设备系统更新维护情况、系统故障统计、通信线路和流量统计、系统验证等多项内容，要对各项情况及时的分析和汇总，并适当建立与前期的对比表格

的 IT 运行环境、业务处理流程、操作规程等发生变化时，要求在灾备中心的灾难备份系统上及时响应并进行同步变更和处理；要做好以上各方面工作，必须在总行数据中心和灾备中心之间建立良好的互动机制。因此，建立一套与之相适应的运营管理制度，对于整个灾难备份项目而言是必不可少的关键工作。

以基于 ITIL 的完善的内部管理制度为基础，数据中心结合客户及委办局实际情况和灾备项目需求，将建立的与之相对应的服务管理接口制度包括的内容如表 9-5 所示。

表 9-5　管理制度

内　　容	说　　明
日常操作运行管理	建立灾难备份系统的日常操作规程，包括： • 灾难备份系统日常监控操作流程 • 灾难备份系统日常操作手册 • 日常维护例行工作流程 • 运行记录及工作报表
事件管理	建立灾难备份系统的问题管理流程，包括： • 事件的受理和记录流程 • 事件定级及知会流程 • 事件追踪及升级流程 • 事件处理结果反馈流程 • 事件的通知策略 • 与变更、问题管理的接口管理
变更管理	建立灾难备份系统的变更管理流程，包括： • 灾难备份系统基准文档维护流程 • 信息系统的变更知会流程 • 信息系统变更评估确认及处理流程 • 业务连续性计划变更维护流程 • 与事件、问题的接口管理
问题管理	建立灾难备份系统的问题管理流程，包括： • 问题的受理和记录流程 • 问题定级及知会流程 • 问题的处理流程 • 问题处理结果反馈流程 • 与事件、变更的接口管理
应急响应及恢复管理	建立灾难备份系统的应急响应管理流程，包括： • 紧急响应流程 • 灾难恢复 IT 及工作环境检查清单
BCP 维护管理	建立 BCP 维护管理流程，包括： • IT 基准维护管理流程 • 子系统验证管理流程 • 灾难恢复预案的分发、保存及版本及更新管理办法
安全管理	建立灾难备份系统安全管理规程，包括： • 安全管理架构 • 备份中心物理安全管理制度 • 安全保密制度 • 网络安全管理流程 • 备份系统分级授权机制 • 磁介质管理制度

9.2.3　运维服务内容综述

根据项目目标的要求以及灾难备份系统的运行特点，数据中心将提供专业的运营管理服务，确保数据中心长期有效的稳定运行。

为此，数据中心提供以下各类专业服务项目：

① 基础设施保障服务；

② 基准验证服务；

③ 现场值守服务；

④ 事件管理服务；

⑤ 问题管理服务；

⑥ 变更管理服务；

⑦ 安全管理服务；

⑧ 客户服务管理；

⑨ 后勤保障服务；

⑩ 运营服务质量管理；

⑪ 服务水平管理；

⑫ 演练配合服务；

⑬ 灾难恢复服务；

⑭ 接替生产中心运行服务。

9.2.4 基础设施保障服务

1. 基础设施保障服务经理

数据中心公司将指定专人作为本项目的基础设施保障服务经理，作为机房基础设施资源保障、维护及相关协调的负责人。

2. 基础设施及机房环境维护

① 供配电系统：数据中心所配备的冗余变压器、高压及低压配电系统、后备柴油发电机、UPS的日常巡检与维护，定期巡检、记录机房配电系统的运行情况，发现问题及时处理。

② 空调系统：数据中心会定时检查机房的环境温湿度，确保机房始终处于符合国家A级机房标准的恒温、恒湿、新风状态，监控空调系统运行情况，发现问题及时处理。

③ 消防系统：数据中心将每月巡检，记录消防系统的运行情况，每年组织进行消防培训和演练。

④ 报警：对机房的UPS、温度、电源等重要环境设施进行集中监控，并实现及时报警。

⑤ 数据中心公司将对数据中心机房的供配电、温湿度环境、漏水、空调、安全等重要设施进行7×24小时集中监控，并实现声光报警。每季度组织厂商对环境监控系统例行维护。

⑥ 数据中心公司将对提供7×24小时保安巡视，每月对门禁系统运行情况进行检查和维护。

⑦ 数据中心公司将对数据中心机房的漏水检测系统每月进行检查测试。

⑧ 数据中心公司将定期检查监控系统，定期对监控录像系统进行维护保养。

⑨ 数据中心公司将提供机房及周边环境的卫生清洁服务。

⑩ 数据中心公司将对所有检查维护行为做详细的记录。

3. 机房安全管理

① 数据中心通过门禁系统控制对机房的访问,非授权人员不得进入。

② 数据中心为机房配备闭路电视监控系统、门禁系统,对机房实行 7×24 小时的实时监控。

③ 对机房实行的闭路电视监控系统、门禁系统等监控记录的磁介质妥善保留一个月。

④ 对机房内所有物品实行严格的事前进出审批及进出登记制度,对记录文档妥善保留一年。

⑤ 提供进出入机房的陪同,并协助满足厂商进行服务的需求。

4. 现场值守服务

(1) 现场支持经理

数据中心公司将由数据中心运营小组组长担任值班经理,负责安排数据中心公司相关人员的值班、巡检、信息汇总等,并及时与客户或委办局相关负责人进行沟通。

(2) 服务内容描述

数据中心公司将提供基础设施环境监控服务,包括供电系统、空调系统、消防系统及安保系统等。7×24 小时定期巡检设备状态,检查设备,并按要求提供巡检记录报告。数据中心公司提供机房 7×24 小时值班服务,值班服务包括机房定时巡查、出入人员登记、配合进行故障排查、简单的连接与断开、介质取放、杂物清理等服务,上述这些服务基于 ITIL(事件,问题和变更的标准流程执行)。

其他服务内容如下。

① 按运行操作手册提供 365×24 小时系统运行操作。

② 填写交班记录和运行日志。

③ 制定数据中心环境设备监控手册, 并按手册要求进行设备监控, 填写监控报告。

④ 按运维管理流程及时向管理系统报告问题/异常。

5. 事件管理服务

事件管理服务的目的是为了尽快恢复基础设施的服务,或响应服务要求。事件管理服务要求记录所有的事件,并建立流程来管理事件的影响。事件管理流程规定了所有事件的记录、优先排序、业务影响、分类、更新、调整、解决和正式关闭,并通知客户,使其了解其报告的事件或服务请求的进展情况,当不能达到约定的服务等级或无法完成约定的措施时应提前警告客户。

① 根据各类用户投诉记录事件: 相关的事件描述、用户信息等。

② 建立知识库,对事件进行初判并试图解决事件。

③ 对事件进行跟踪,并能及时通知相关人员尽快解决事件, 直到事件结束。

④ 对未按时解决的事件,进入事件升级流程,即将事件升级并上报上级主管。

⑤ 协调相关人员快速解决事件。

⑥ 提供事件查询及事件进程查询。

⑦ 对各类变更及事件影响程度按流程进行通知。

⑧ 评估相关人员解决事件所需时间,进行绩效考核。

⑨ 系统能够对相关维护人员进行每月的统计故障解决率、故障解决及时率。

⑩ 统计每月、每年的事件次数、故障次数。

事件管理所涉及的所有人员都可以访问相关的信息，如已知错误、事件解决方案和配置管理数据库。应对重大事件进行分类并根据过程进行管理。

事件管理包括确定、记录、跟踪和纠正项目实施过程中出现的事件，并制定相应的解决方案以降低或预防事件的重复出现。

一般地，事件等级分为一级事件、二级事件、三级事件、四级事件。

6. 事件管理上报

对于项目执行阶段，数据中心公司和客户（或委办局）所定义的一、二级事件真正发生时，数据中心公司提供如下上报管理服务使客户（或委办局）时刻了解事件的进展和解决情况。

严重等级一、二级的事件的通知策略（将根据项目具体调整）如表 9-6 和表 9-7 所示。

表 9-6　严重等级一、二级的事件的通知策略

严重等级	灾备中心	事件负责人	事件管理员	甲方项目经理	事件负责人—客户服务经理(PM)
1	联系灾备中心报告事件	灾备中心立即电话通知	灾备中心立即电话通知	灾备中心发送 SMS/Email 通知（小于 15 分钟）和客户服务经理电话通知（小于 30 分钟）	灾备中心发送 SMS/Email 通知（小于 15 分钟）
2	联系灾备中心报告事件	灾备中心立即电话通知	灾备中心立即电话通知	灾备中心发送 SMS/Email 通知（小于 25 分钟）和客户经理电话通知（小于 30 分钟）	灾备中心发送 SMS/Email 通知（小于 15 分钟）
3	联系灾备中心报告事件	灾备中心立即电话通知	N/A	N/A	N/A
4	联系灾备中心报告事件	灾备中心立即电话通知	N/A	N/A	N/A

表 9-7　严重等级一、二级的事件的通知策略

事件管理	基础设施（HVAC）事件管理		
	一级严重级别事件	在 30 分钟内做出响应，在 2 个小时内解决	
	二级严重级别事件	在 60 分钟内做出响应，在 4 个小时内解决	
	三级严重级别事件	在 4 小时内做出响应，在 48 个小时内解决	
	四级严重级别事件	在 24 小时内做出响应，在一周内解决	
	事件情况统计和记录报告	每月提交，提交时间不迟于下月服务周期开始后 10 个工作日内	

9.2.5　变更管理服务

变更管理服务是 IT 服务管理标准 ITIL 中的一个重要范畴。数据中心在多年的灾难备份服务经历中深刻体会到：在对灾难备份系统的长期的服务管理工作中，变更管理是一个极其关键的、绝对不可忽视的重要工作范畴。可以毫不夸张地说，变更管理的好坏，将直接影响到我们建设的灾难备份系统，是否能够在紧急灾难关头成功地实现系统及时切换并恢复运行。

根据数据中心对灾难备份系统多年维护管理的经验，我们认为灾难备份系统通常具有以下的特性。

① 灾难备份系统一旦建立，需要在长达数年甚至数十年间的整个服务周期内一直保持其完整性和有效可用性。

② 灾难备份系统所面对的生产系统一定会变更，灾难备份系统的变更将和生产系统的变更保持一致或处理性能的匹配，但生产系统的变更并不依赖于灾难备份系统的变更完成。

③ 对灾难备份系统的变更需求通常会被忽略，使灾难备份系统的变更经常处于未知或被动的状态。

因此，当灾难备份系统建立并投入运行以后，为切实地保证这个备份系统能够长期有效地与生产系统保持同步可用，一个关键问题就是要保证备份系统根据生产系统的变更保持有效同步的变更。

灾备中心根据在国内灾难恢复行业多年运作的经验，结合国内客户的特点制定出一整套切实可靠和行之有效的变更管理服务措施，能够在长期维护客户灾难备份系统的服务工作中，确保客户灾难备份系统根据生产系统的变更进行有效同步的互动，并在工作中尽力保证生产系统和备份系统的平稳变更。

1. 变更管理目标

数据中心将提供流程化的变更管理，减少或避免因为疏忽、缺少资源、准备不充分等缘故导致变更失败或产生其他的问题。

① 确保所有对灾难备份系统的变更都遵循标准的方法、程序和规则进行。

② 确保所有对灾难备份系统的变更都能快捷有效地进行。

③ 减少或避免与变更相关的事故对灾难备份系统运行的影响。

④ 确保所有对灾难备份系统的变更都有明确、完整的记录可追踪。

⑤ 确保所有的变更都有严格审核机制和恢复方案。

⑥ 通过对变更的评估管理，采取相应的控制措施控制变更的风险。

2. 变更内容分析

灾难备份系统所面临的变更范围将涵盖硬件、软件、网络、通信、服务要求、文档流程等几个方面。

(1) 硬件变更

由于生产系统硬件配置的变更而引起的对灾难备份系统的设备硬件进行的变更，比如硬件更换、硬件升级等。

(2) 软件变更

由于生产系统的操作系统或相关软件的变更所引起的对灾难备份系统中相关系统软件、应用软件、通信软件等进行的变更，比如操作系统版本升级、软件产品版本升级或更换、镜像卷组的调整等。

(3) 网络变更

由于生产网络配置的变更所引起的对灾难备份系统网络配置进行的变更，比如路由配置、网点配置等。

(4) 通信变更

为满足灾难备份系统对通信线路需求而进行的变更，比如带宽升级等。

(5) 服务要求变更

如对 mirror view 数据复制系统暂停/恢复的要求,对备份磁带循环的周期变更等。

(6) 文档流程变更

对支持或维护灾难备份系统正常运行的所有相关文档、规程、流程进行的变更。

此外,根据灾难备份系统的运行特点,灾难备份系统发生的变更将分为以下几种可能的类型。举例如下。

(1) 常规变更:对灾难备份系统所进行的变更是双方预先约定的变更范围,相应的变更工作流程事先应已经双方共同确定。

(2) 项目变更:对灾难备份系统所需进行的变更是双方无法事先约定的,在灾难备份系统长期运行过程中随着业务发展而产生的,这类变更往往涉及备份系统的配置变更,需要经过双方进行变更方案及成本的协商确定后方可实施的变更。比如设备硬件升级、通信线路带宽升级、服务范围增加等。

(3) 紧急变更:由于意外原因引发的对灾难备份系统进行的临时变更,需要提交对方紧急处理,如通信线路临时发送维修暂停通知等情况所产生的变更。

3. 变更管理服务方式

根据以上变更内容分析,灾备中心提供的变更管理服务将从以下几个方面实现。

(1) 建立有效的变更互动机制

数据中心可根据对灾难备份系统维护服务的经验,提出一套变更知会的通知及互动流程,在此基础上,双方再针对生产系统的运行和管理特点,共同协商和制定一套可共同操作的《变更知会单》及变更处理流程。该变更处理流程同时适用于同城和异地灾备中心。

(2) 变更响应热线

灾备中心将提供 7×24 小时客户服务热线电话和应急响应电话,可在任何时间内响应和受理客户提交的变更请求,将变更请求及时分派到有关部门进行相应处理之后,还将跟踪变更的执行情况,将处理结果及时反馈或以书面汇报。

(3) 定期基准核对

灾备中心将定期进行灾难备份系统的基准环境核对工作,对有关设备的硬件配置、操作系统版本/配置、应用软件版本、网络设备及配置等情况进行统计与核对,同时根据统计核对的结果更新双方的文档资料,确保灾难备份系统与生产系统处于同步更新的状态之中。

(4) 定期统计报告

灾备中心将定期提供当期灾难备份系统的变更汇总统计报告,以便客户(或委办局)详细了解灾难备份系统的变更情况及变更后的现状。

4. 变更处理流程

常规变更和紧急变更的发起源自于两个不同方面,为此,数据中心灾难备份中心初步考虑了以下两类变更处理流程框架,具体变更处理流程将在新一期项目实施中进行完善。

(1) 重大变更处理流程

当因业务发展需要,在服务期内对生产数据中心进行同城范围内的搬迁或对灾难备份系统现有的设备、系统、网络、服务方案等提出变更需求时,数据中心将对变更需求所涉及的范围进行评估,在确定服务内容不变的前提下,根据评估的结果提交变更实施报告,包括变更的实施计划、实施方案及所需的资源、测试的目标等,双方就变更报告共同商定完善并予

以实施。

(2) 紧急变更处理流程

紧急变更实施过程中存在很大的不确定性,应尽量减少紧急变更,当紧急变更无法避免时,按以下流程进行处理:灾备中心、客户或委办局任何一方提出紧急变更请求时,都应该根据影响度、紧迫性、优先级对变更进行评估,确定变更请求属于紧急变更,否则按常规变更处理流程进行处理。确认变更为紧急变更后,双方必须通过双方约定的方式通知对方,并共同进行紧急变更实施方案准备,如有时间对变更实施方案进行测试,应尽量安排测试,确保变更实施方案的安全性、可操作性。双方根据确定后的紧急变更实施方案协调相关资源实施变更。若变更实施成功,总结变更实施情况,并知会相关人员紧急变更处理完成;否则启动回退计划,重新准备变更实施方案,组织变更实施。

9.2.6 问题管理服务

问题管理服务也是 IT 服务管理标准 ITIL 中的一个极为重要的范畴。根据数据中心在灾难备份服务行业的实践经验,我们认识到:面向灾难备份系统的运行特点及维护要求,问题管理是一个需要认真对待和认真管理的重要工作范畴。问题管理的水平,将直接影响到以高科技手段建立起来的灾难备份系统,是否能够在日复一日、年复一年的长期运行过程中保持稳定可靠的运转,从而确保灾难备份系统真正能够达到我们预期的标准和希望。

1. 问题管理目标

数据中心根据自身多年的数据中心管理以及灾难恢复运作经验,结合国内客户的特点,制定了一套完善有效的问题管理服务制度和措施。长期以来,数据中心在维护客户灾难备份系统的工作中,正是依靠这套制度做到了:在发生问题前,能尽早消除隐患;在发生问题后,能及时发现问题,并有效控制问题影响范围;在定位问题后,能够分析问题发生的原因,迅速解决问题,并有效防止同类问题的再次发生。

数据中心将提供流程化的问题管理,减少或避免因人为疏忽或处理不当等原因而导致的任何过失。

① 确保所有对灾难备份系统的问题处理都按照标准的流程和规则有序进行。

② 确保所有对灾难备份系统的问题处理都能快捷有效地进行。

③ 确保所有对灾难备份系统的问题处理都有明确、完整的记录可追踪。

④ 对所有问题的处理过程和结果都要进行事后评估,采取相应的控制措施来避免问题的再次发生。

2. 问题管理服务方式

数据中心灾难备份系统提供的变更管理服务将从以下几个方面实现。

(1) 严密防备

灾备中心将根据客户或委办局灾难备份系统的生产运行要求,建立相关的设备、系统、网络巡检、监控机制,并严格执行。

(2) 实时监控

灾备中心将为客户或委办局灾难备份系统提供 7×24 小时的运行监控服务,以确保可以及时发现和报告问题。

(3) 问题响应热线

灾备中心将提供7×24小时服务热线电话,可在任何时间内响应和受理问题报告,将问题报告及时分派到有关部门进行相应处理之后,还将跟踪问题的执行情况,将处理结果向相关单位及时反馈或以书面形式汇报。

(4) 问题响应处理

灾备中心将配备7×24小时值班的技术工程师,确保可以及时响应、处理灾难备份系统发生的问题。

(5) 问题处理汇报

灾备中心对每次问题发生及处理的结果进行经验和教训的总结,根据需要更新双方的灾难恢复文档资料,确保灾难备份系统的持续稳定运行。

(6) 定期统计报告

灾备中心将定期提供当期灾难备份系统的问题汇总统计报告,对灾难备份系统发生的问题进行趋势分析,发现灾难备份系统存在的隐患,提出处理建议,以便相关各单位详细了解灾难备份系统的运行情况及发展趋势。

3. 问题管理流程

问题的发现及报告将会源自于两个不同方面,为此,数据中心初步考虑了以下两类问题处理流程框架,具体问题处理流程还将在新一期项目实施中进行完善。

(1) 对发起的问题报告的处理流程

当从生产中心一端发现灾难备份系统发生问题时,可按下述处理流程向灾备中心通报问题,双方将尽快解决问题,以求尽可能降低问题对灾难备份系统的影响或尽快恢复灾难备份系统的正常运行,如图9-5所示。

以上流程的主要处理环节如下。

详细记录问题发生的时间、涉及的设备系统、问题现象,并按约定的方式及时通知灾备中心。灾备中心接到发来的问题通知后,值班工程师将立即到位,并组织力量对问题进行处理,尽快恢复灾难备份系统的正常运行。问题处理完成后,灾备中心将提交相关问题处理报告。若问题需要客户或委办局进行处理,相关单位应协调相关资源进行问题处理,灾备中心值班工程师将全力配合进行问题处理。问题处理完成后,相关单位应提交相关问题处理报告。

(2) 对数据中心灾难备份中心发起的问题报告的处理流程

当灾备中心在灾难备份系统的监控过程中发现灾难备份系统运行出现问题时,灾备中心将按下述处理流程报告问题,双方将尽快解决问题,以求尽可能降低问题对灾难备份系统的影响或尽快恢复灾难备份系统的正常运行。主要处理环节如下。

灾备中心将对问题作详细的记录,并及时报告数据中心灾难备份中心值班工程师。问题记录的内容包括:①问题发生的时间和日期;②问题现象描述;③问题发现人的信息。

灾备中心值班工程师受理问题后,将按以下方式对问题进行归类、定级,并按约定的方式及时通知相关人员。

① 确定与问题相关的服务;

② 问题对服务的影响情况;

③ 问题的大小、范围和复杂程度;

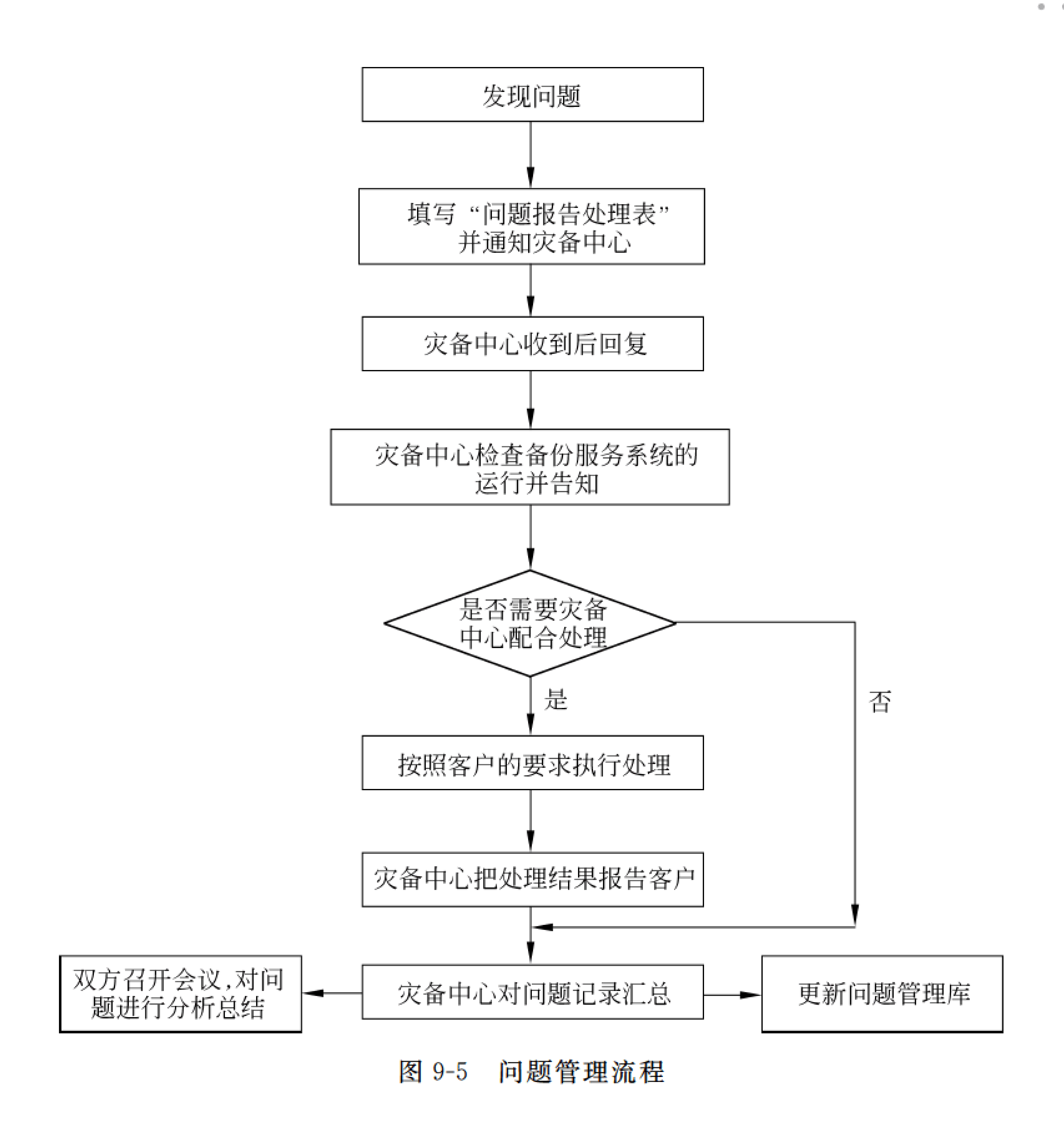


图 9-5　问题管理流程

④ 当前可供处理事故的资源。

灾备中心将协调资源，包括第三方厂商，对问题进行调查分析，找出问题发生的根本原因，提出问题的解决方案，并组织对问题解决方案进行测试。若问题解决方案测试不通过，将重新提交问题解决方案。

数据中心协调相关资源实施问题解决方案，解决问题，恢复灾难备份系统的正常运行。

问题处理完成后，灾备中心将提交问题处理报告，对问题处理情况进行总结，必要时提出预防措施，防止问题的再次发生。

若问题需要客户或委办局进行处理，相关单位应协调相关资源进行问题处理，数据中心灾难备份中心值班工程师将全力配合问题的处理。问题处理完成后，相关单位应提交相关问题处理报告。

(3) 重大问题处理流程

当灾备中心或客户(委办局)在对灾难备份系统的监控过程中发现灾难备份系统运行出现重大问题时，双方应本着快速报告、快速解决的精神对重大问题进行处理和解决，以求尽可能降低问题对灾难备份系统的影响或尽快恢复灾难备份系统的正常运行。

9.2.7　客户服务管理

为确保数据中心提供的灾难备份服务符合约定要求，灾备中心除了建立以上的各项管

理服务体系之外，还要确保客户对灾备中心服务质量及对灾难备份系统的运行情况的评价及时准确地反馈到灾备中心。

为此，数据中心对提供的各项服务进行客户服务水平的质量管理控制。

在以客户为中心的客户服务水平管理过程中，数据中心会从以下几个方面切实体现真正优质的服务管理水平。

1. 完善服务标准

数据中心将根据客户灾难备份系统的运行特点，在服务中达到灾备中心应提供的灾难备份运营服务指标要求，并以此为目标来严格执行。

2. 建立联动机制

建立生产中心与同城灾难备份中心联动机制，包括联络方式、对应人员等，一旦出现紧急情况，根据约定在第一时间联系对应人员共同解决问题，提高事件响应速度。

3. 客户经理负责

数据中心将指定专门的客户经理，定期前往客户或委办局征询对灾难备份中心服务情况的反映，并将问题和建议及时反馈到数据中心，敦促公司各有关部门和灾难备份中心做出及时的改进和修正，并将改进措施及结果及时向相关单位汇报。

4. 月度服务报告

数据中心将每月定期向服务客户提交上月的系统运行月报，其中将包含以下各项内容：

① 机房运行情况汇报；

② 数据复制系统的运行情况、数据统计及维护情况汇报；

③ 备份设备系统的检查维护情况汇报；

④ 灾难备份系统的变更汇总情况汇报；

⑤ 问题处理情况汇报；

⑥ 服务水平(SLA)执行情况汇报；

⑦ 存在问题与解决建议；

⑧ 双方往来文档列表；

⑨ 其他约定的情况汇报。

5. 定期评估总结

数据中心将定期召开灾难备份服务情况的评估总结会。总结会将由数据中心提交当期的服务运行报告，针对各项服务情况进行汇报，并由服务客户对整体服务情况进行评估总结，最后双方共同对存在的差异制定相关的改进措施，在不断总结的基础上持续进步。

9.2.8 运维服务质量管理

为保证灾备系统安全、稳定、有效地运行，运行维护系统服务质量起到关键作用，数据中心公司将制定一套对服务质量的管理评估标准，并按照服务合约对服务的质量进行检查和评估。

1. 服务质量管理

为确保数据中心公司提供的服务是可测量、可评估、可持续改进的专业服务。数据中心根据服务水平协议和工作内容说明制定服务质量测评管理标准，对生产系统维护服务过程的服务目标进行定义，对服务范围明确说明、服务指标量化、服务质量要求与测评方法进行

定义,然后定期进行检查,并定期按此标准进行评测和检查。

2. 服务质量评估

数据中心对日常运营服务的内容进行量化定义,定期提交系统运行的服务水平达成表,总结分析服务提供情况以及对服务存在的差异进行服务质量的评估。

9.2.9　服务水平管理

1. 客户服务经理

数据中心公司将指定专人作为本项目的客户服务经理。客户服务经理作为客户和数据中心公司的唯一正式桥梁,沟通,协调和管理数据中心公司内部的一切资源是为了提供优运维管理外包服务。

客户服务经理将组织运维服务管理团队并对数据中心公司投标书中提供的所有服务情况进行监控和自我检查。并且与相关人员及时沟通,了解最新需求并及时反馈。

2. 服务内容描述

服务水平管理是保证数据中心公司能够按照合同约定的服务内容和服务水平顺利交付服务的最基础保证。此服务的提供者是数据中心公司客户服务经理。

数据中心公司的客户服务管理遵循数据中心公司多年外包经验总结的 GPMM(数据中心程序管理模式)项目管理方法论,针对这套方法论,数据中心公司将对运营管理提供项目管理服务。

数据中心客户服务经理每月根据需要和项目经理组织召开双方的服务总结例会,总结所在机房基础设施维护的执行情况和相关厂商的服务情况。

9.2.10　灾难恢复服务

当生产中心发生灾难时,灾备中心提供如下灾难恢复服务。

(1) 系统网络准备就绪检查包括以下方面。

① 灾备中心系统运行状况检查;

② 灾备中心网络设备运行状况检查;

③ 灾备中心与生产中心、各支机构通信线路、网络流量及传输质量检查;

④ 机房环境准备就绪检查;

⑤ 特殊授权处理流程启动,包括灾备中心人员出入、机房门禁卡、用户使用等。

(2) 环境及支持人员准备就绪检查。

(3) 协助信息管理服务包括以下方面。

① 提供灾备中心整体运行状况报告;

② 灾备系统切换准备就绪报告;

③ 协助通知相关人员及单位。

(4) 场地保障支持服务。

(5) 网络保障支持服务及所需网络开通服务。

(6) 配合执行灾难恢复预案包括以下三点。

① 实施系统切换;

② 进度控制;

③ 实施中问题记录、分析、总结及报告。

（7）后勤保障准备就绪。

9.2.11 应急资源保障

1. 人员保障

人力资源的保障是应急保障措施中的一项重要工作内容。在应急响应工作中，各部门应服从应急领导组、应急管理组的统一协调和安排。

应加强信息系统突发事件应急技术支持队伍的建设，通过各种应急管理与应急技术的培训和应急演练，不断提高应急人员的业务素质、技术水平和应急处置能力。

2. 设备保障

应按需要配备相应的应急设备。应急设备包括但不限于以下方面：

① 重要信息系统主机的冗余配置以及灾备；

② 通信网络的关键设备部件（核心路由器、核心交换机等的冗余配置）；

③ 应急响应相关工具，包括木马、后门检测工具等。

对于重要信息系统的设备部件，根据应急情况需要临时配备应急设备时，可与相关厂商签署紧急供货协议或租赁协议。

3. 财务保障

财务部门应对信息系统应急响应所需的相关经费给予充分保障，相关经费需求应包括但不局限以下几个方面：

① 应急响应相关人员培训费用；

② 应急设备购置费；

③ 应急演练相关费用；

④ 应急支援服务费等。

4. 通信保障

应急期间，指挥、通信联络和信息交换的渠道应该得到保证，主要通信方式包括但不限于电话、手机、传真、电子邮件等，有关应急联系的手机应保持24小时开机状态。

应急期间，需要按照事先规定的沟通路径在信息系统应急响应组织成员之间进行沟通，以保证通信渠道畅通，避免信息通道拥堵。

5. 后勤保障

信息管理部门应做好信息系统应急响应后勤保障工作，确保信息系统应急响应工作的顺利开展。

信息系统应急响应后勤保障主要包括：应急人员食宿、交通的安排，应急物资的运输，应急办公环境的提供和管理。

6. 技术保障

应急技术资料是成都市工商局信息系统重要的技术信息，包括网络拓扑结构、应用系统（包含数据库、中间件、应用软件）的配置、相关设备的型号及配置、主要服务商信息等详细信息。市局信息管理部门应将这些信息建立技术档案并及时更新，以保证与实际情况的一致性。

在经费得以保障的情况下，应与厂商和服务商签订服务水平协议，加入应急事件技术支

持的内容，保证外部技术人员在应急事件发生时能够在规定时间内到达现场，参与事件损害评估、系统抢修及系统切换等工作。

市局信息管理部门应与第三方的技术专家保持沟通，逐步建立应对各种信息系统突发事件的应急专家组。

9.2.12　应急宣传、培训与演练

1. 宣传

市局信息管理部门应利用各种途径宣传应急响应相关法律法规和应急响应基础知识，开展信息系统突发事件应急预防、预警、自救、互救和减灾等知识的宣传活动，普及信息系统应急处置的基本知识。

2. 培训

在网络与信息系统安全应急预案编制完成和修订后，市局信息管理部门应通过培训使有关人员熟练掌握应急处理的程序，明确自己在突发事件中所承担的角色和职责，提高应急处理能力。

为确保网络和信息系统安全应急预案有效运行，市局信息管理部门、各单位在网络与信息系统安全应急预案编制完成和修订后，应定期或不定期组织不同层次、不同类型的培训班或研讨会，以便不同岗位的应急人员都能全面熟悉并掌握信息系统应急处理的知识和技能。

涉及预案的各级人员应结合本岗位安全职责和应急预案的要求，管理人员应熟练掌握本单位应急预案中有关报警、接警、处警和组织指挥应急响应的程序等内容，专项应急预案操作人员应熟悉各个操作步骤和操作命令。

各单位网络与信息系统安全教育应包括本单位应急预案的有关内容，使有关人员熟悉本单位应急处理的流程、应急处理设施的使用、应急联系电话、应急报告的内容和格式。

3. 演练

为了验证、完善和优化预案中的各项内容（包括管理组织机构、人员安排、应急流程、资源保障等），确保在突发事件发生的情况下，具备对突发事件的应急响应处理能力，制定预案的演练及维护计划。

演练的主要目的在于确认当前应急恢复所制定的策略、组织机构、人员角色、应急响应具体流程均已被所有相关人员所充分了解，验证现有应急流程的正确性和有效性，发现潜在的问题，完善和优化现有的应急流程，符合预期目标。

演练采取循序渐进的方法，在范围上遵循由小到大的原则；在流程上采用分阶段的处理方法；在演练的过程中不断测试和完善应急工作中的各个环节，最终达到对整个应急流程的测试和完善。

应急预案的演练需要制订详细周全的演练计划及相关准备工作，合理安排，精细组织，确保演练工作的安全，并制定出各种意外情况出现时的应急措施。

要明确演练的目的和要求，记录演练过程，对演练结果进行评估和总结。

各单位应根据信息系统的关键点和薄弱点，根据系统和设备的重要程度有针对性地开展演练，演练应突出重点和关键。

各专项应急预案制定后，各单位要组织相应的演练，在重大节假日前均应开展相关的演

练。各单位每年应至少组织一次联合演习。

各单位要通过演练验证本单位应急预案和各专项应急预案的合理性，及时修订和完善应急预案和各专项应急预案。

9.3 安全管理服务

为实现灾备系统运营管理中可靠性的目标，数据中心将提供对于灾备中心以及灾难备份系统的安全控制。在项目实施准备期间，数据中心将与客户共同开发制订信息安全管理文件，并就有关项目安全控制实施得到双方共同认可。数据中心须严格遵守双方签订的灾难备份服务系统相关合同中的保密条款。

9.3.1 安全管理通则

数据中心提供以下安全管理通则，并承担相应的责任。

（1）数据中心确认其在中国境内开展灾难备份系统相关运营服务的合法合规性。

（2）确认满足国家和行业主管部门的从业资质要求。

（3）在项目实施准备期间，与客户或委办局确认日常运行手册中安全管理内容，并在双方共同认可后予以实施。

（4）数据中心将依据国际通行的 ISO9000、ISO20000 与 ISO27001 等标准，在灾备中心内部建立 QMS 与 ISMS，强化灾备系统的信息安全管理。

（5）数据中心将通过内外部审核工作的开展，确保各项制度、流程、工作得到合理的实施。

（6）数据中心将根据法律、法规、技术与客户的要求，定期调整灾备中心的安全管理体系，确保安全管理的要求能真正落实到灾备中心的日常运营当中。

9.3.2 人员的安全

灾备中心确认其在中国境内开展灾难备份系统相关运营服务的合法合规性，灾难备份中心将严格遵守国家的法律、法规，确保在本运营服务期间涉及的客户数据安全和业务秘密。

（1）所有数据中心参与本运营服务项目的人员，均与公司签订保密协议。

（2）灾备中心及其员工承诺将严格遵守法律、法规，决不将客户的业务数据、客户信息透露给第三方（依法披露除外）。

（3）灾备中心将严格限制和控制自己的职员，并及时获得客户的业务信息（包括各种应用和业务数据、报表、传真、文档等）。

（4）灾备中心将对访问项目敏感信息的职员和其他人员进行必要的培训，保证其拥有适当的技能和资质，保证其掌握必要的信息安全管理制度和安全保密意识。

（5）灾备中心将对员工实施严格的考核机制与聘用管理，从员工进入灾备中心开始到离开灾备中心，进行全程监控管理。

9.3.3 物理的安全

(1) 为确保灾备中心的公共安全,数据中心与附近的公安联防、消防部门等建立密切联系。

(2) 灾备中心采用有效的全区域实时监控和周界红外线报警系统及 110 联网报警系统,对园区实行集中监控、7×24 小时保安巡逻,全面保证中心园区物理设施的安全性。

(3) 灾备中心对机房等关键区域采用 7×24 小时的监控录像,录像数据保留一年。

(4) 灾备中心园区实行严格的授权准入制度与分区域管理制度,外来人员需获得授权、并在内部人员陪同下,才能进入园区的各安全管制区域。

(5) 灾备中心关键区域和机房实行严格的门禁系统保护的准入制度。

(6) 灾备中心对需进出园区或机房的设备和物品履行严格的核查及放行手续,其中进出机房的设备还必须获得中心管理层的审批方可核查与放行。

9.3.4 安全审核

(1) 数据中心将通过外审服务的选择,从第三方的角度去检查灾备中心的安全管理水平。

(2) 数据中心在灾备中心内部成立专门的内审机构,定期对中心运营工作进行审核,确保中心的工作能符合法律、法规、合约以及客户的相关要求。

(3) 数据中心会配合客户聘请或指派的外审人员,完成对灾备系统管理的符合性检查。

任务拓展

1. 请简要介绍运维体系。
2. 为什么我们需要安全运维?
3. 安全运维需要注意哪些方面?



第四篇

数据灾备趋势

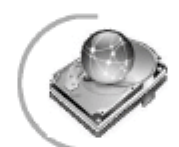


第 10 章　云计算应用与云灾备



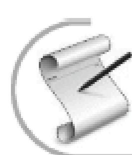
任务目标

- 了解云计算的应用；
- 了解云的服务模式和模型；
- 了解云灾备。



项目背景

“云技术”已经越来越多地进入人们的工作与生活。专家预测，未来 5 年，互联网和通信方面 90%的新业务将来自于云技术。



项目描述

什么是云？为什么云技术如此高效？我们应该怎么去使用云？云在灾备系统中扮演了一个什么样的角色？本章将会就以上问题为大家进行解答。



项目分析

通过学习和了解云技术，为将来的云架构搭建、使用、服务储备基础知识，提高对云计算的认识。



项目实现

通过对本章内容的学习，了解云灾备的架构和服务方式。

10.1　云计算的应用

10.1.1　云物联

“物联网就是物物相连的互联网”。这有两层意思：第一，物联网的核心和基础仍然是互联网，是在互联网基础上的延伸和扩展的网络；第二，其用户端延伸和扩展到了任何物品与物品之间进行信息交换和通信。

物联网的两种业务模式如下。

(1) MAI(M2M Application Integration)，即内部 MaaS。

(2) MaaS(M2M As A Service)，MMO，Multi-Tenants(多租户模型)。

随着物联网业务量的增加，对数据存储和计算量的需求将带来对“云计算”能力的要求。

(1) 云计算：从计算中心到数据中心在物联网的初级阶段，PoP 即可满足需求。

(2) 在物联网高级阶段，可能出现 MVNO/MMO 营运商(国外已存在多年)，需要虚拟化云计算技术与 SOA 等技术的结合实现互联网的泛在服务，即 TaaS(Every Thing as a Service)。

10.1.2 云安全

云安全(Cloud Security)是一个从“云计算”演变而来的新名词。云安全的策略构想是：使用者越多，每个使用者就越安全，因为如此庞大的用户群，足以覆盖互联网的每个角落，只要某个网站被挂马或某个新木马病毒出现，就会立刻被截获。

“云安全”通过网状的大量客户端对网络中软件行为的异常监测，获取互联网中木马、恶意程序的最新信息，推送到 Server 端进行自动分析和处理，再把病毒和木马的解决方案分发到每一个客户端。

下面介绍云安全的 10 种方法。

1. 密码优先

通常状态下，任何网站或服务器的用户名和密码只有一个，而且要得到许可。理由很简单：如果用户名和密码都是同一组，那么当其中一个被盗了，其他的账户也暴露了。

2. 检查安全问题

在设置访问权限时，尽可能绕开一些简单易明白的问题，例如 QQ 头像。最好的方法是选择一个问题，而这个问题的答案却是另一个问题的答案。例如，如果你选择的问题是“小时候住在哪里”，答案却是“黄色”之类的不关联内容。

3. 试用加密方法

不管这样的方法效果如何，不可否认，也是一个不错的想法。加密软件需要来自用户方面的努力，但它也有可能需要你去抢夺代码凭证，加大使用成本。

4. 管理密码

不言而喻，你有很多的用户名和密码需要管理，为了管理它们，要有应用程序和软件来辅助你工作。

5. 双重认证

在允许用户访问网站之前可能会有两种使用模式。因此除了用户名和密码之外，唯一验证码也是必不可少的。这一验证码可能是以短信的形式发送到你的手机上，然后进行登录。这种方式的优势在于，即使你的凭证丢了，但是别人没有你的验证码，也是登录不上的。

6. 不要犹豫，立刻备份

当涉及云中数据保护时，人们被告知在物理硬盘上要进行数据备份，这听起来可能有些奇怪，但这确实是需要你去做的事。这就是为什么需要一遍一遍反复思考。你应该直接在外部硬盘上备份数据，并随身携带。

7. 完成即删除

为什么有无限的数据存储选择时，我们还要费时间去做删除工作呢？原因在于，你永远不知道有多少数据有潜在的危险。如果来自于某家银行账户的邮件或警告信息时间太长，已经失去了价值，那么就删除它。

8. 注意登录的信息

很多情况下，我们都会在别人的设备上登录，但是我们通常会忘了这样就有可能会保留自己的信息，从而存在暴露的风险。

9. 使用反病毒、反间谍软件

尽管是云数据，但使用这一方法的原因在于你第一次从系统中访问云。因此，如果系统存在风险，那么你的在线数据也将存在风险。一旦你忘记加密，那么键盘监听就会获得你的云厂商密码，最终你将失去所有。

10. 时刻都要管住自己的嘴巴

永远都不要把你的云存储内容与别人共享。保持密码的秘密性是必需的。为了附加的保护功能，不要告诉别人你使用的所有厂商。

10.1.3 云存储

云存储是在云计算(Cloud Computing)概念上延伸和发展出来的一个新的概念，是指通过集群应用、网格技术或分布式文件系统等功能，将网络中大量各种不同类型的存储设备通过应用软件集合起来协同工作，共同对外提供数据存储和业务访问功能的一个系统。当云计算系统运算和处理的核心是大量数据的存储和管理时，云计算系统中就需要配置大量的存储设备，那么云计算系统就转变成为一个云存储系统，所以云存储是一个以数据存储和管理为核心的云计算系统。

10.1.4 云游戏

云游戏是以云计算为基础的游戏方式，在云游戏的运行模式下，所有游戏都在服务器端运行，并将渲染完毕后的游戏画面压缩后通过网络传送给用户。在客户端，用户的游戏设备不需要任何高端处理器和显卡，只需要基本的视频解压能力就可以了。就现今来说，云游戏并没有成为家用机和掌机界的联网模式。但是在若干年以后，云计算取代这些东西成为其网络发展的终极方向的可能性很大。如果这种构想能成为现实，那么主机厂商将变成网络运营商，他们不需要不断投入巨额的新主机研发费用，而只需要拿这笔钱中的很小一部分去升级自己的服务器就行了，但是达到的效果却是相差无几的。对于用户来说，他们可以省下购买主机的开支，但是得到的却是顶尖的游戏画面(当然视频输出方面的硬件必须过硬)。

10.1.5 云计算

从技术上看，大数据与云计算的关系就像一枚硬币的正面和反面一样密不可分。大数据必然无法用单台的计算机进行处理，必须采用分布式计算架构。它的特色在于对海量数据的挖掘，但它必须依托云计算的分布式处理、分布式数据库、云存储。

10.1.6 私有云

私有云(Private Cloud)是将云基础设施与软硬件资源创建在防火墙内，以供机构或企业内各部门共享数据中心内的资源。创建私有云，除了硬件资源外，一般还有云设备(IaaS)软件。现时商业软件有 VMware 的 vSphere 和 Platform Computing 的 ISF。开放源代码的云设备软件主要有 Eucalyptus 和 OpenStack。

10.1.7 云教育

视频云计算应用在教育行业的实例：流媒体平台采用分布式架构部署，分为 Web 服务器、数据库服务器、直播服务器和流服务器，如有必要可在信息中心架设采集工作站搭建网络电视或实况直播应用，在各个学校已经部署录播系统或直播系统的教室配置流媒体功能组件，这样录播实况可以实时传送到流媒体平台管理中心的全局直播服务器上，同时录播的学校本色课件也可以上传存储到金山区教育局信息中心的流存储服务器上，方便今后的检索、点播、评估等各种应用。

10.1.8 云会议

云会议是基于云计算技术的一种高效、便捷、低成本的会议形式。使用者只需要通过互联网界面进行简单易用的操作，便可快速高效地与全球各地团队及客户同步分享语音、数据文件及视频，而会议中数据的传输、处理等复杂技术由云会议服务商帮助使用者进行操作。

目前国内云会议主要集中在以 SAAS(软件即服务)模式为主体的服务内容，包括电话、网络、视频等服务形式，基于云计算的视频会议就叫云会议。云会议是视频会议与云计算的完美结合，带来了最便捷的远程会议体验。及时与移动云电话会议，是云计算技术与移动互联网技术的完美融合，可以通过移动终端进行简单的操作，从而可以随时随地高效地召集和管理会议。

10.1.9 云社交

云社交(Cloud Social)是一种物联网、云计算和移动互联网交互应用的虚拟社交应用模式，以建立著名的“资源分享关系图谱”为目的，进而开展网络社交，云社交的主要特征，就是把大量的社会资源统一整合和评测，构成一个资源有效池，向用户按需提供服务。参与分享的用户越多，能够创造的利用价值就越大。

10.2 云的三种服务模式和四种服务模型

10.2.1 云服务的模式

云计算可以认为包括以下几个层次的服务：基础设施级服务(IaaS)，平台级服务(PaaS)和软件级服务(SaaS)。这里所谓的层次，是分层体系架构意义上的“层次”。IaaS、PaaS、SaaS 分别在基础设施层、软件开放运行平台层和应用软件层实现。

1. IaaS

IaaS(Infrastructure as a Service)：基础设施即服务，消费者通过 Internet 可以从完善的计算机基础设施获得服务。IaaS 是把数据中心、基础设施等硬件资源通过 Web 分配给用户的商业模式。

2. PaaS

PaaS(Platform as a Service)：平台即服务。PaaS 实际上是指将软件研发的平台作为一种服务，以 SaaS 的模式提交给用户。因此，PaaS 也是 SaaS 模式的一种应用。但是，PaaS

的出现可以加快 SaaS 的发展，尤其是加快 SaaS 应用的开发速度。PaaS 服务使得软件开发人员可以在不购买服务器等设备环境的情况下开发新的应用程序。

3. SaaS

SaaS(Software as a Service)：软件即服务。它是一种通过 Internet 提供软件的模式，用户无须购买软件，而是向提供商租用基于 Web 的软件来管理企业经营活动。

SaaS 模式大大降低了软件，尤其是大型软件的使用成本，并且由于软件是托管在服务商的服务器上降低了客户的管理维护成本，可靠性也更高。

10.2.2　云服务的部署模型

1. 公有云

公有云通常指第三方提供商为用户提供的能够使用的云。公有云一般可通过 Internet 使用，可能是免费或成本低廉的。公有云的核心属性是共享资源服务。这种云有许多实例，可在当今整个开放的公有网络中提供服务。例如，中国移动、中国联通、中国电信、IDC 运营商、亚马逊、IBM、Joyent、Rackspaces 等。

在此种模式下，应用程序、资源、存储和其他服务，都由云服务供应商来提供给用户，这些服务大部分都是免费的，也有部分按需按使用量来付费，这种模式只能使用互联网来访问和使用。同时，这种模式在私人信息和数据保护方面也比较有保证。这种部署模型通常都可以提供可扩展的云服务并能高效设置。

2. 私有云

私有云(Private Clouds)是为一个客户单独使用而构建的，因而提供对数据、安全性和服务质量的最有效控制。该公司拥有基础设施，并可以控制在此基础设施上部署应用程序的方式。私有云可部署在企业数据中心的防火墙内，也可以将它们部署在一个安全的主机托管场所，私有云的核心属性是专有资源。

这种云基础设施专门为某一个企业服务，不管是自己管理还是第三方管理，自己负责还是第三方托管都可以。只要使用的方式没有问题，就能为企业带来很显著的帮助。

不过这种模式所要面临的纠正、检查等安全问题则需企业自己负责，否则出了问题也只能自己承担后果，此外，整套系统也需要自己出钱购买、建设和管理。这种云计算模式可产生非常广泛的正面效益，从模式的名称也可以看出，它可以为所有者提供具备充分优势和功能的服务。

私有云的特点如下。

(1) 数据安全

虽然每个公有云的提供商都对外宣称，其服务在各方面都是非常安全，特别是对数据的管理。但是对企业而言，特别是大型企业而言，与业务有关的数据是其生命线，不能受到任何形式的威胁，所以短期而言，大型企业是不会将其 Mission-Critical 的应用放到公有云上运行的。而私有云在这方面是非常有优势的，因为它一般都构筑在防火墙后。

(2) SLA(服务质量)

因为私有云一般在防火墙之后，而不是在某一个遥远的数据中心内，所以当公司员工访问那些基于私有云的应用时，它的 SLA 应该会非常稳定，不会受到网络不稳定的影响，比如上次“暴风影音”事件就导致了大规模的断网。

（3）充分利用现有硬件资源和软件资源

大家知道许多大公司都会有很多 legacy 的应用，而且 legacy 大多都是其核心应用。虽然公有云的技术很先进，但对 legacy 应用的支持并不理想，因为这些应用很多都是用静态语言编写的，以 Cobol、C、C++ 和 Java 为主，而现有的公有云对这些语言支持很一般。但私有云在这方面就不错，比如 IBM 推出的 CloudBurst，通过 CloudBurst，能非常方便地构建基于 Java 的私有云。而且一些私有云的工具能够利用企业现有的硬件资源来构建云，这样将极大降低企业的成本。

（4）不影响现有 IT 管理的流程

对大型企业而言，流程是其管理的核心，如果没有完善的流程，企业将会成为一盘散沙。不仅与业务有关的流程非常繁多，而且 IT 部门的流程也不少，比如那些和 Sarbanes-Oxley 相关的流程，并且这些流程对 IT 部门非常关键。在这一方面，公有云会很吃亏，因为假如使用公有云，将会对 IT 部门流程有很多的冲击，比如在数据管理方面和安全规定等方面。而私有云因为一般在防火墙内，所以对 IT 部门流程冲击不大。

3. 社区云

“社区云”是大的“公有云”范畴内的一个组成部分，是指在一定的地域范围内，由云计算服务提供商统一提供计算资源、网络资源、软件和服务能力所形成的云计算形式。即基于社区内的网络互联优势和技术易于整合等特点，通过对区域内各种计算能力进行统一服务形式的整合，结合社区内的用户需求共性，实现面向区域用户需求的云计算服务模式。

社区云是一些由有着类似需求并打算共享基础设施的组织共同创立的云，社区云的目的是实现云计算的一些优势。由于社区云的用户数比公有云少，这种选择往往比公有云贵，但隐私度安全性和政策遵从都比公有云高。

“社区云会是大的云计算的互联世界里非常富有活力的组成部分，我们可以把它生动地理解为‘云朵’。每一个云朵都基于云计算技术实现，实现了资源的共享，服务的统一，但同时每一个云朵都具有自己鲜明的特征，比如区域特色，也可能是行业特点。‘社区云’通过更大范围的互联，成为云计算世界里的组成部分。基于‘社区云’的先进架构设计，结合下一代互联网的便利，我们会非常便利地提取出‘云朵’的优势服务，为更大范围内的相似用户提供服务。”这段由冉雨先生在 2011 年初的深圳 YOCSEF 研讨会上的发言，生动地描述了“社区云”的概念、特点和未来的发展方向。

社区云具有如下特点：

① 区域型和行业性；

② 有限的特色应用；

③ 资源的高效共享；

④ 社区内成员的高度参与性。

4. 混合云

混合云融合了公有云和私有云，是近年来云计算的主要模式和发展方向。我们已经知道私有云主要是面向企业用户，出于安全考虑，企业更愿意将数据存放在私有云中，但是同时又希望可以获得公有云的计算资源，在这种情况下混合云被越来越多的企业采用，它将公有云和私有云进行混合和匹配，以获得最佳的效果，这种个性化的解决方案，达到了既省钱又安全的目的。

混合云是两种或两种以上的云计算模式的混合体，如公有云和私有云混合。它们相互独立，但在云的内部又相互结合，可以发挥出所混合的多种云计算模型各自的优势。

混合云特点如下。

(1) 更完美

私有云的安全性是超越公有云的，而公有云的计算资源又是私有云无法企及的。在这种矛盾的情况下，混合云完美地解决了这个问题，它既可以利用私有云的安全，将内部重要数据保存在本地数据中心；同时也可以使用公有云的计算资源，更高效快捷地完成工作，相比私有云或是公有云都更完美。

(2) 可扩展

混合云突破了私有云的硬件限制，利用公有云的可扩展性，可以随时获取更高的计算能力。企业通过把非机密功能移动到公有云区域，可以降低对内部私有云的压力和需求。

(3) 更节省

混合云可以有效地降低成本。它既可以使用公有云又可以使用私有云，企业可以将应用程序和数据放在最适合的平台上，获得最佳的利益组合。

10.3　云灾备介绍

10.3.1　云灾备的概念

云灾备将灾备看作是一种服务，由客户付费使用灾备服务，提供商提供灾备的服务模式。采用这种模式，客户可以利用服务提供商的优势技术资源、丰富的灾备项目经验和成熟的运维管理流程，快速实现用户的灾备目标，降低客户的运维成本和工作强度，同时也降低灾备系统的总体拥有成本。

10.3.2　云灾备服务详情

云灾备服务是采用当前最先进、安全、可靠的数据备份和数据复制技术，建设可管理、可运营的灾备服务，为企事业单位、政府部门提供不同等级的同城或异地灾备服务，以保证在灾难发生后能够快速、准确地恢复客户的业务数据和关键应用系统，保障客户业务的连续运行。此外，灾备服务可进一步降低企事业单位、政府部门的信息化成本，并为政府层面监督管理和宏观决策提供平台和工具支撑。

10.3.3　云灾备的服务类型

1. 数据级灾备

数据级灾备的关注点在于数据，即灾难发生后，灾备服务平台依靠基于网络的数据复制工具，实现生产中心和灾备中心之间的异步/同步的数据传输，可以确保客户原有的业务数据不遭破坏。

2. 应用级容灾

在数据级灾备的基础上构建应用级灾备系统，具备应用系统接管能力，即在异地灾备中心再构建一套支撑系统、备用网络系统等部分。当生产环境发生故障时，灾备中心可以接管

应用继续运行,减少系统宕机时间,保证业务的连续性。

10.3.4　云灾备服务优势

（1）建设机制：从建设简化为采购。

（2）业务实现难易度：按需采购,即购即用,系统可靠运行。

（3）可升级和可迁移性：按需调整,随时可以升级或迁移,随着企业的发展设备可以方便地降级重复利用,大大节约系统升级费。

（4）成本问题：购买即可使用,运行成本低,维护简单,而且有运营商和厂商的服务保障。

（5）任何用户都可采用。

10.3.5　基于云灾备的数据安全存储关键技术

使用云存储系统提供的第三方云灾备服务以其成本低、弹性架构、低接入门槛的特点逐渐得到政府、企业以及个人用户的关注。但是由于云灾备服务的安全性和第三方的可信性问题,目前的云存储仅开展了个人用户的在线备份服务,目前多个地、市、州政府已经开始筹建能提供海量数据灾备的服务。总的来说,基于云灾备服务的研究主要面临以下问题。

（1）数据机密性与重复数据删除的矛盾

目前的云灾备服务为了保证多用户间重复数据删除的效率,用户和云存储系统间传输和存储的都是明文数据,因此第三方环境中数据的机密性是云灾备数据安全存储中用户最为关心的一个问题,即使用户对数据进行了加密处理,存储在第三方透明环境中的数据将无法有效地进行数据重删,尤其当系统中存储的为高冗余度的灾备数据存储空间利用率将无法得到保障,因此,数据的机密性与云灾备服务的存储空间利用率的矛盾将成为云灾备服务发展面临的最主要矛盾。

（2）缺乏全存储周期的完整性检查手段

云灾备系统的不透明性增大了用户对于数据完整性检查的需求。但是目前的云存储系统数据完整性检查都集中关注于用户的云存储系统交互时的数据完整性检查,在数据存储阶段缺乏有效的完整性检查方法。同时对于数据完整性检查的另一个重点,即数据的非授权的改动方面,目前仅仅使用存储系统中的权管理手段进行,无法有效地隔离共享式存储环境中不同用户的数据访问以及恶意用户或系统管理员针对数据的非法访问。

（3）缺乏应用层的数据可用性保护手段

数据恢复是业务连续性计划最重要的一环,尽管在传统的存储环境中对于可生存存储技术和容错技术的研究已经十分成熟,但是使用云存储系统搭建的数据容灾系统中数据的可用性保护使用最多的仍然是数据副本冗余技术。同时,在云存储环境下的数据安全存储关键技术的可用性,尽在低层次的物理设备层次进行数据冗余,缺乏更高层次的能够让用户信任的数据可用性技术。

（4）缺乏统一数据安全管理手段

当前的云存储系统为用户提供数据存储服务时,数据的机密性、完整性、可用性通过不同的数据安全策略来保证。这些策略分别使用不同的参数和不同的算法对数据进行处理,目前没有一个统一的基于云灾备的数据安全存储模型能够同时兼顾数据机密性、完整性、可

用性。同时，相关的技术研究缺乏从用户和云灾备服务商两者对于安全性的需求角度进行构造的灾备数据安全存储架构。

10.4　Google 云计算原理介绍

Google 拥有全球最强大的搜索引擎。除了搜索业务以外，Google 还有 Google Maps、Google Earth、Gmail、YouTube 等各种业务，包括刚诞生的 Google Wave。这些应用的共性在于数据量巨大，而且要面向全球用户提供实时服务，因此 Google 必须解决海量数据存储和快速处理问题。Google 的诀窍在于它发展出简单而又高效的技术，让多达百万台的廉价计算机协同工作，共同完成这些前所未有的任务，这些技术是在诞生几年之后才被命名为 Google 云计算技术。Google 云计算技术具体包括：Google 文件系统 GFS、分布式计算编程模型 MapReduce、分布式锁服务 Chubby 和分布式结构化数据存储系统 Bigtable 等。其中，GFS 提供了海量数据的存储和访问的能力，MapReduce 使得海量信息的并行处理变得简单易行，Chubby 保证了分布式环境下并发操作的同步问题，Bigtable 使得海量数据的管理和组织十分方便。本节对这四种核心技术进行详细介绍。

10.4.1　Google 文件系统 GFS

Google 文件系统（Google File System，GFS）是一个大型的分布式文件系统。它为 Google 云计算提供海量存储，并且与 Chubby、MapReduce 以及 Bigtable 等技术结合十分紧密，处于所有核心技术的底层。由于 GFS 并不是一个开源的系统，我们仅仅能从 Google 公布的技术文档来获得一点了解，而无法进行深入的研究。

当前主流分布式文件系统有 RedHat 的 GFS(Global File System)、IBM 的 GPFS、Sun 的 Lustre 等。这些系统通常用于高性能计算或大型数据中心，对硬件设施要求较高。以 Lustre 文件系统为例，它只对元数据管理器 MDS 提供容错解决方案，而对于具体的数据存储节点 OST 来说，则依赖其自身来解决容错的问题。例如，Lustre 推荐 OST 节点采用 RAID 技术或 SAN 存储区域网来容错，但由于 Lustre 自身不能提供数据存储的容错，一旦 OST 发生故障就无法恢复，因此对 OST 的稳定性就提出了相当高的要求，从而大大增加了存储的成本，而且成本会随着规模的扩大线性增长。

正如李开复所说的那样，创新固然重要，但有用的创新更重要。创新的价值取决于一项创新在新颖、有用和可行性这三个方面的综合表现。Google GFS 的新颖之处并不在于它采用了多么令人惊讶的技术，而在于它采用廉价的商用机器构建分布式文件系统，同时将 GFS 的设计与 Google 应用的特点紧密结合，并简化其实现，使之可行，最终达到创意新颖、有用、可行的完美组合。GFS 使用廉价的商用机器构建分布式文件系统，将容错的任务交由文件系统来完成，利用软件的方法解决系统可靠性问题，这样可以使得存储的成本成倍下降。由于 GFS 中服务器数目众多，在 GFS 中服务器死机是经常发生的事情，甚至都不应当将其视为异常现象，那么如何在频繁的故障中确保数据存储的安全、保证提供不间断的数据存储服务是 GFS 最核心的问题。GFS 的精彩在于它采用了多种方法，从多个角度并使用不同的容错措施来确保整个系统的可靠性。

10.4.2 系统架构

GFS 的系统架构如图 10-1 所示，GFS 将整个系统的节点分为三类角色：Client（客户端）、Master（主服务器）和 Chunk Server（数据块服务器）。Client 是 GFS 提供给应用程序的访问接口，它是一组专用接口，不遵守 POSIX 规范，以库文件的形式提供。应用程序直接调用这些库函数，并与该库链接在一起。Master 是 GFS 的管理节点，在逻辑上只有一个，它保存系统的元数据，负责整个文件系统的管理，是 GFS 文件系统中的大脑。Chunk Server 负责具体的存储工作。数据以文件的形式存储在 Chunk Server 上，Chunk Server 的个数可以有多个，它的数目直接决定了 GFS 的规模。GFS 将文件按照固定大小进行分块，默认是 64MB，每一块称为一个 Chunk（数据块），每个 Chunk 都有一个对应的索引号（Index）。

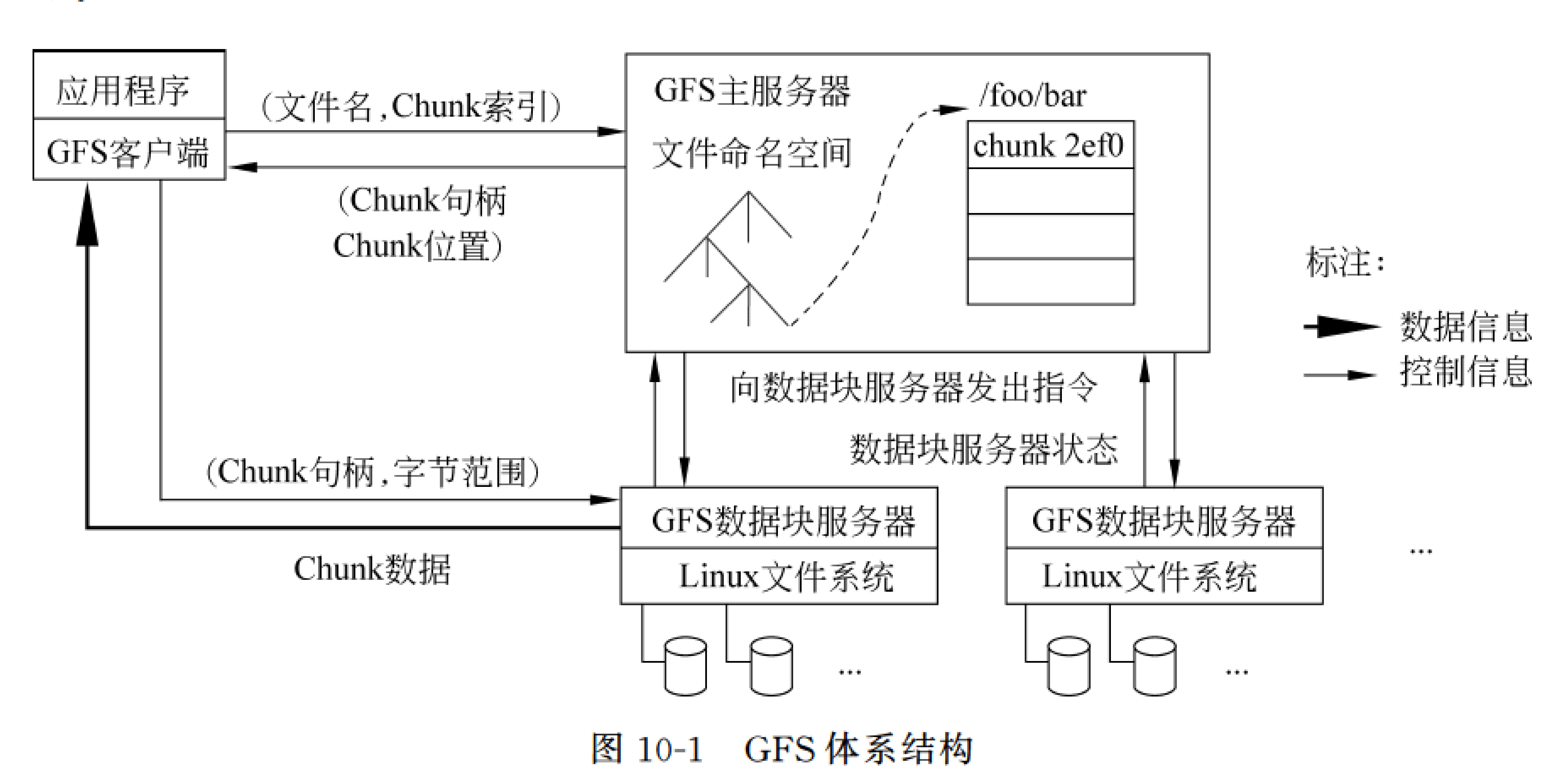


图 10-1　GFS 体系结构

客户端在访问 GFS 时，首先访问 Master 节点，获取将要与之进行交互的 Chunk Server 信息，然后直接访问这些 Chunk Server 并完成数据的存取。GFS 的这种设计方法实现了控制流和数据流的分离。Client 与 Master 之间只有控制流，而无数据流，这样就极大地降低了 Master 的负载，使之不成为系统性能的一个瓶颈。Client 与 Chunk Server 之间直接传输数据流，同时由于文件被分成多个 Chunk 进行分布式存储，Client 可以同时访问多个 Chunk Server，从而使得整个系统 I/O 高度并行，系统整体性能得到提高。

相对于传统的分布式文件系统，GFS 针对 Google 应用的特点从多个方面进行了简化，从而在一定规模下达到成本、可靠性和性能的最佳平衡。具体来说，它具有以下几个特点。

1. 采用中心服务器模式

GFS 采用中心服务器模式来管理整个文件系统，可以大大简化设计，从而降低实现难度。Master 管理了分布式文件系统中的所有元数据。文件划分为 Chunk 进行存储，对于 Master 来说，每个 Chunk Server 只是一个存储空间。Client 发起的所有操作都需要先通过 Master 才能执行。这样做有许多好处，增加新的 Chunk Server 是一件十分容易的事情，Chunk Server 只需要注册到 Master 上即可，Chunk Server 之间无任何关系。如果采用完全对等的、无中心的模式，那么如何将 Chunk Server 的更新信息通知到每一个 Chunk Server，会是设计的一个难点，而这也将在一定程度上影响系统的扩展性。Master 维护了一

个统一的命名空间，同时掌握了整个系统内 Chunk Server 的情况，据此可以实现整个系统范围内数据存储的负载均衡。由于只有一个中心服务器，元数据的一致性问题自然解决。当然，中心服务器模式也带来一些固有的缺点，比如极易成为整个系统的瓶颈等。GFS 采用多种机制来避免 Master 成为系统性能和可靠性上的瓶颈，如尽量控制元数据的规模、对 Master 进行远程备份、控制信息和数据分流等。

2. 不缓存数据

缓存机制是提升文件系统性能的一个重要手段，通用文件系统为了提高性能，一般需要实现复杂的缓存(Cache)机制。GFS 文件系统根据应用的特点，没有实现缓存，这是从必要性和可行性两方面考虑的。从必要性上讲，客户端大部分是流式顺序读写，并不存在大量的重复读写，缓存这部分数据对系统整体性能的提高作用不大；而对于 Chunk Server，由于 GFS 的数据在 Chunk Server 上以文件的形式存储，如果对某块数据读取频繁，本地的文件系统自然会将其缓存。从可行性上讲，如何维护缓存与实际数据之间的一致性是一个极其复杂的问题，在 GFS 中各个 Chunk Server 的稳定性都无法确保，加之网络等多种不确定因素，一致性问题尤为复杂。此外由于读取的数据量巨大，以当前的内存容量无法完全缓存。对于存储在 Master 中的元数据，GFS 采取了缓存策略，GFS 中 Client 发起的所有操作都需要先经过 Master。Master 需要对其元数据进行频繁操作，为了提高操作的效率，Master 的元数据都是直接保存在内存中进行操作；同时采用相应的压缩机制降低元数据占用空间的大小，提高内存的利用率。

3. 在用户态下实现

文件系统作为操作系统的重要组成部分，其实现通常位于操作系统底层。以 Linux 为例，无论是本地文件系统如 Ext3 文件系统，还是分布式文件系统如 Lustre 等，都是在内核态实现的。在内核态实现文件系统，可以更好地和操作系统本身结合，向上提供兼容的 POSIX 接口。然而，GFS 却选择在用户态下实现，主要基于以下考虑。

(1) 在用户态下实现，直接利用操作系统提供的 POSIX 编程接口就可以存取数据，无须了解操作系统的内部实现机制和接口，从而降低了实现的难度，并提高了通用性。

(2) POSIX 接口提供的功能更为丰富，在实现过程中可以利用更多的特性，而不像内核编程那样受限。

(3) 用户态下有多种调试工具，而在内核态中调试相对比较困难。

(4) 用户态下，Master 和 Chunk Server 都以进程的方式运行，单个进程不会影响到整个操作系统，从而可以对其进行充分优化。在内核态下，如果不能很好地掌握其特性，效率不但不会高，甚至还会影响到整个系统运行的稳定性。

(5) 用户态下，GFS 和操作系统运行在不同的空间，两者耦合性降低，从而方便 GFS 自身和内核的单独升级。

4. 只提供专用接口

通常的分布式文件系统一般都会提供一组与 POSIX 规范兼容的接口。其优点是应用程序可以通过操作系统的统一接口来透明地访问文件系统，而不需要重新编译程序。GFS 在设计之初是完全面向 Google 的应用的，采用了专用的文件系统访问接口。接口以库文件的形式提供，应用程序与库文件一起编译，Google 应用程序在代码中通过调用这些库文件的 API，完成对 GFS 文件系统的访问。采用专用接口有以下好处。

（1）降低了实现的难度。通常与POSIX兼容的接口需要在操作系统内核一级实现，而GFS是在应用层实现的。

（2）采用专用接口可以根据应用的特点提供一些特殊支持，如支持多个文件并发追加的接口等。

（3）专用接口直接和Client、Master、Chunk Server交互，减少了操作系统之间上下文的切换，降低了复杂度，提高了效率。

10.4.3 容错机制

1. Master容错

具体来说，Master上保存了GFS文件系统的三种元数据。

（1）命名空间（Name Space），也就是整个文件系统的目录结构。

（2）Chunk与文件名的映射表。

（3）Chunk副本的位置信息，每一个Chunk默认有三个副本。

首先就单个Master来说，对于前两种元数据，GFS通过操作日志来提供容错功能。第三种元数据信息则直接保存在各个Chunk Server上，当Master启动或Chunk Server向Master注册时自动生成。因此当Master发生故障时，在磁盘数据保存完好的情况下，可以迅速恢复以上元数据。为了防止Master彻底死机的情况，GFS还提供了Master远程的实时备份，这样在当前的GFS Master出现故障无法工作的时候，另外一台GFS Master可以迅速接替其工作。

2. Chunk Server容错

GFS采用副本的方式实现Chunk Server的容错。每一个Chunk有多个存储副本（默认为三个），分布存储在不同的Chunk Server上。副本的分布策略需要考虑多种因素，如网络的拓扑、机架的分布、磁盘的利用率等。对于每一个Chunk，必须将所有的副本全部写入成功，才视为成功写入。在其后的过程中，如果相关的副本出现丢失或不可恢复等状况，Master会自动将该副本复制到其他Chunk Server，从而确保副本保持一定的个数。尽管一份数据需要存储三份，好像磁盘空间的利用率不高，但综合比较多种因素，加之磁盘的成本不断下降，采用副本无疑是最简单、最可靠、最有效，而且实现的难度也是最小的一种方法。

GFS中的每一个文件被划分成多个Chunk，Chunk的默认大小是64MB，这是因为Google应用中处理的文件都比较大，以64MB为单位进行划分，是一个较为合理的选择。Chunk Server存储的是Chunk的副本，副本以文件的形式进行存储。每一个Chunk以Block为单位进行划分，大小为64KB，每一个Block对应一个32bit的校验和。当读取一个Chunk副本时，Chunk Server会将读取的数据和校验来进行比较，如果不匹配，就会返回错误，从而使Client选择其他Chunk Server上的副本。

10.4.4 系统管理技术

严格意义上来说，GFS是一个分布式文件系统，包含从硬件到软件的整套解决方案。除了上面提到的GFS的一些关键技术外，还有相应的系统管理技术来支持整个GFS的应用，这些技术可能并不一定为GFS所独有。

1. 大规模集群安装技术

安装 GFS 的集群中通常有非常多的节点，最大的集群超过 1000 个节点，而现在的 Google 数据中心动辄有万台以上的机器在运行。那么，迅速地安装、部署一个 GFS 的系统，以及迅速地进行节点的系统升级等，都需要相应的技术支撑。

2. 故障检测技术

GFS 是构建在不可靠的廉价计算机之上的文件系统，由于节点数目众多，故障发生十分频繁，如何在最短的时间内发现并确定发生故障的 Chunk Server，需要相关的集群监控技术。

3. 节点动态加入技术

当有新的 Chunk Server 加入时，如果需要事先安装好系统，那么系统扩展将是一件十分烦琐的事情。如果能够做到只需将裸机加入，就会自动获取系统并安装运行，那么将会大大减少 GFS 维护的工作量。

4. 节能技术

有关数据表明，服务器的耗电成本大于当初的购买成本，因此 Google 采用了多种机制来降低服务器的能耗，例如对服务器主板进行修改，采用蓄电池代替昂贵的 UPS(不间断电源系统)，提高能量的利用率。Rich Miller 在一篇关于数据中心的博客文章中表示，这个设计让 Google 的 UPS 利用率达到 99.9%，而一般数据中心只能达到 92%～95%。

10.4.5　并行数据处理 MapReduce

MapReduce 是 Google 提出的一个软件架构，是一种处理海量数据的并行编程模式，用于大规模数据集(通常大于 1TB)的并行运算。Map(映射)、Reduce(化简)的概念和主要思想，都是从函数式编程语言和矢量编程语言借鉴来的。正是由于 MapReduce 有函数式和矢量编程语言的共性，使得这种编程模式特别适合于非结构化和结构化的海量数据的搜索、挖掘、分析与机器智能学习等。

1. 产生背景

MapReduce 这种并行编程模式思想最早是在 1995 年提出的。与传统的分布式程序设计相比，MapReduce 封装了并行处理、容错处理、本地化计算、负载均衡等细节，还提供了一个简单而强大的接口。通过这个接口，可以把大尺度的计算自动地并发和分布执行，从而使编程变得非常容易。还可以通过由普通 PC 构成的巨大集群来达到极高的性能。另外，MapReduce 也具有较好的通用性，大量不同的问题都可以简单地通过 MapReduce 来解决。

MapReduce 把对数据集的大规模操作，分发给一个主节点管理下的各分节点共同完成，通过这种方式实现任务的可靠执行与容错机制。在每个时间周期，主节点都会对分节点的工作状态进行标记，一旦分节点状态标记为死亡状态，则这个节点的所有任务都将分配给其他分节点重新执行。

据相关统计，每使用一次 Google 搜索引擎，Google 的后台服务器就要进行 10^{11} 次运算。这么庞大的运算量，如果没有好的负载均衡机制，有些服务器的利用率会很低，有些则会负荷太重，有些甚至可能死机，这些都会影响系统对用户的服务质量。而使用

MapReduce 这种编程模式，就保持了服务器之间的均衡，提高了整体效率。

2. 编程模型

MapReduce 的运行模型如图 10-2 所示。图中有 M 个 Map 操作和 R 个 Reduce 操作。

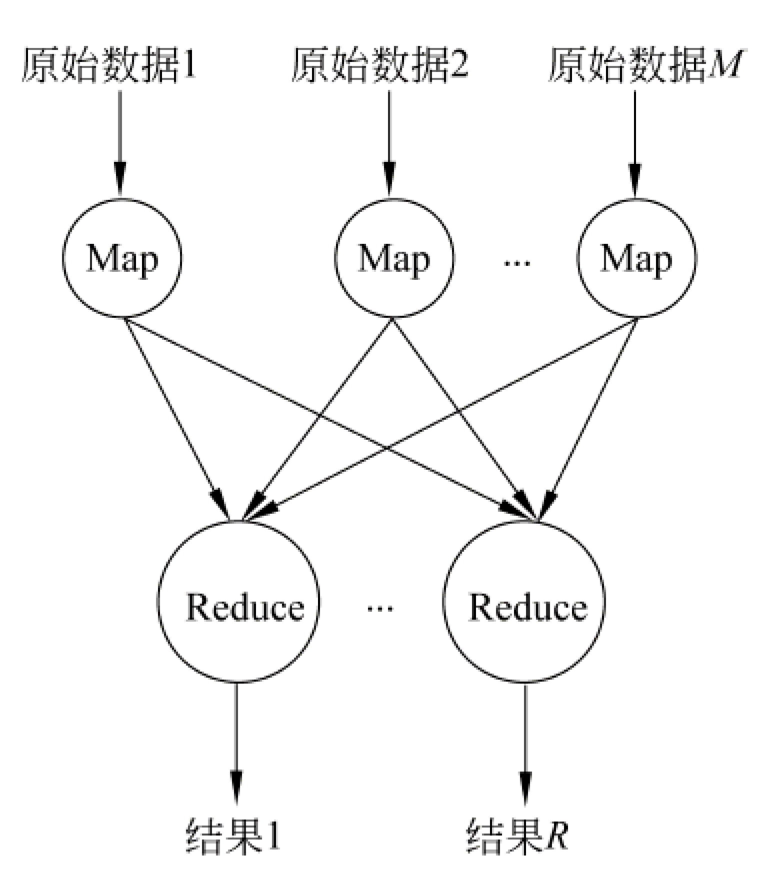


图 10-2　MapReduce 的运行模型

简单地说，一个 Map 函数就是对一部分原始数据进行指定的操作。每个 Map 操作都针对不同的原始数据，因此 Map 与 Map 之间是互相独立的，这就使得它们可以充分并行化。一个 Reduce 操作就是对每个 Map 所产生的一部分中间结果进行合并操作，每个 Reduce 所处理的 Map 中间结果是互不交叉的，所有 Reduce 产生的最终结果经过简单连接就形成了完整的结果集，因此 Reduce 也可以在并行环境下执行。

在编程的时候，开发者需要编写两个主要函数：

① Map：(in_key, in_value)→{(key$_j$, value$_j$)|j=1…k}

② Reduce：(key, [value$_1$,…,value$_m$])→(key, final_value)

Map 和 Reduce 的输入参数和输出结果根据应用的不同而有所不同。Map 的输入参数是 in_key 和 in_value，它指明了 Map 需要处理的原始数据是哪些。Map 的输出结果是一组＜key，value＞对，这是经过 Map 操作后所产生的中间结果。在进行 Reduce 操作之前，系统已经将所有 Map 产生的中间结果进行了归类处理，使得相同 key 对应的一系列 value 能够集结在一起提供给一个 Reduce 进行归并处理，也就是说，Reduce 的输入参数是(key，[value$_1$,…,value$_m$])。Reduce 的工作是需要对这些对应相同 key 的 value 值进行归并处理，最终形成(key, final_value)的结果。这样，一个 Reduce 处理了一个 key，所有 Reduce 的结果并在一起就是最终结果。

例如，假设我们想用 MapReduce 来计算一个大型文本文件中各个单词出现的次数，Map 的输入参数指明了需要处理哪部分数据，以＜在文本中的起始位置，需要处理的数据长度＞表示，经过 Map 处理，形成一批中间结果＜单词，出现次数＞。而 Reduce 函数则是把中间结果进行处理，将相同单词出现的次数进行累加，得到每个单词总的出现次数。

10.4.6　实现机制

实现 MapReduce 操作的执行流程如图 10-3 所示。

当用户程序调用 MapReduce 函数，就会引起如下操作(图中的数字标示和下面的数字标示相同)。

用户程序中的 MapReduce 函数库首先把输入文件分成 *M* 块，每块大概 16～64MB(可以通过参数决定)，接着在集群的机器上执行处理程序。

这些分派的执行程序中有一个程序比较特别，它是主控程序 Master。剩下的执行程序都是作为 Master 分派工作的 Worker(工作机)。总共有 *M* 个 Map 任务和 *R* 个 Reduce 任务需要分派，Master 选择空闲的 Worker 来分配这些 Map 或者 Reduce 任务。

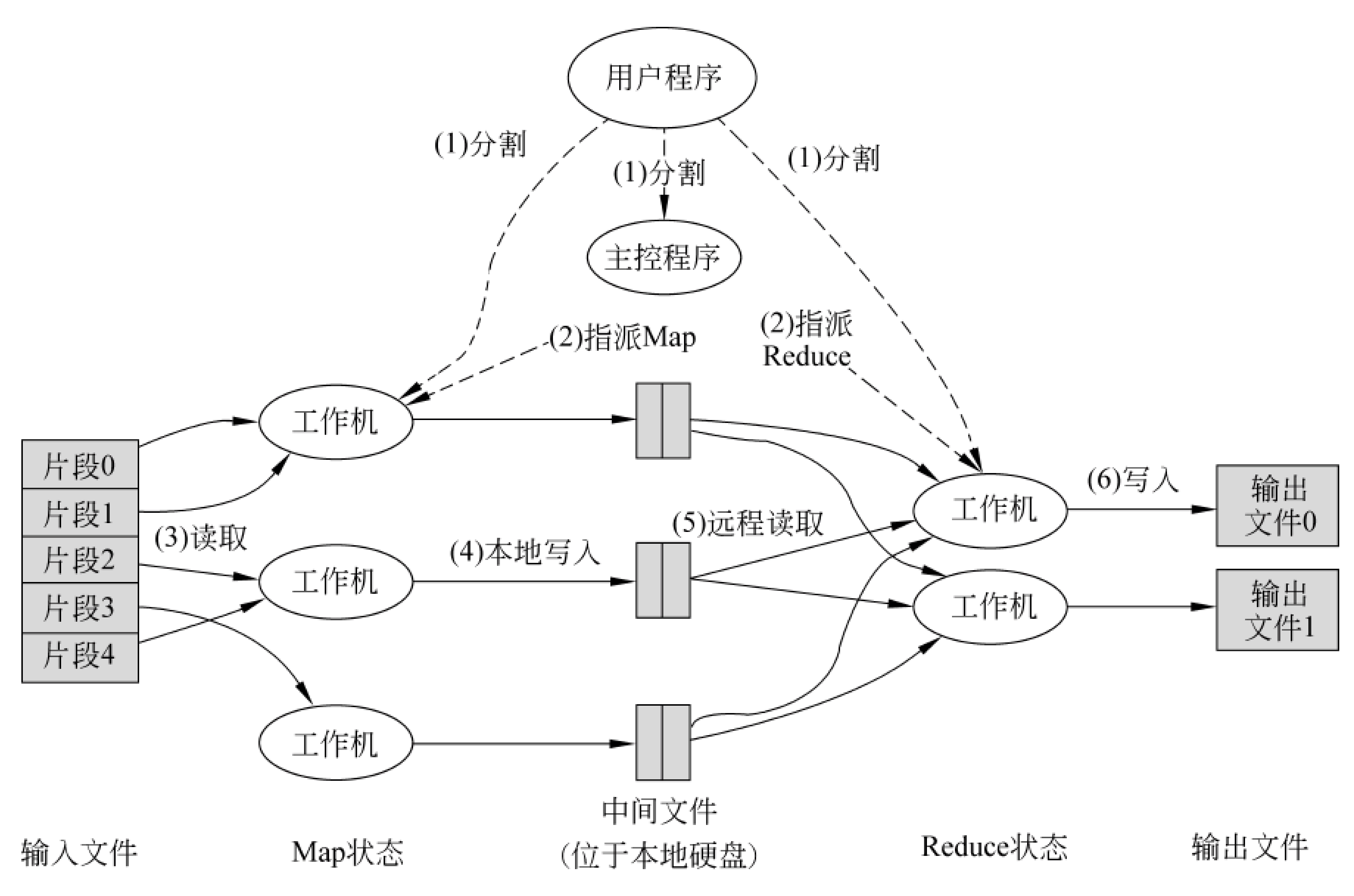


图 10-3　MapReduce 执行流程图

一个分配了 Map 任务的 Worker 读取并处理相关的输入块。它处理输入的数据，并且将分析出的＜key，value＞对传递给用户定义的 Map 函数。Map 函数产生的中间结果＜key，value＞对暂时缓冲到内存。

这些缓冲到内存的中间结果将被定时写到本地硬盘，这些数据通过分区函数分成 R 个区。中间结果在本地硬盘的位置信息将被发送回 Master，然后 Master 负责把这些位置信息传送给 Reduce Worker。

当 Master 通知 Reduce 的 Worker 关于中间＜key，value＞对的位置时，它调用远程过程来从 Map Worker 的本地硬盘上读取缓冲的中间数据。当 Reduce Worker 读到所有的中间数据，它就使用中间 key 进行排序，这样可以使得相同 key 的值都在一起。因为有许多不同 key 的 Map 都对应相同的 Reduce 任务，所以，排序是必需的。如果中间结果集过于庞大，那么就需要使用外排序。

Reduce Worker 根据每一个唯一中间 key 来遍历所有的排序后的中间数据，并且把 key 和相关的中间结果值集合传递给用户定义的 Reduce 函数。Reduce 函数的结果输出到一个最终的输出文件中。

当所有的 Map 任务和 Reduce 任务都已经完成的时候，Master 激活用户程序。此时 MapReduce 返回用户程序的调用点。

由于 MapReduce 是用在成百上千台机器上处理海量数据的，所以容错机制是不可或缺的。总的说来，MapReduce 是通过重新执行失效的地方来实现容错的。

1. Master 失效

在 Master 中，会周期性地设置检查点（checkpoint），并导出 Master 的数据。一旦某个任务失效了，就可以从最近的一个检查点恢复并重新执行。不过由于只有一个 Master 在运行，如果 Master 失效了，则只能终止整个 MapReduce 程序的运行并重新开始。

2. Worker 失效

相对于 Master 失效而言，Worker 失效算是一种常见的状态。Master 会周期性地给 Worker 发送 ping 命令，如果没有 Worker 的应答，则 Master 认为 Worker 失效，终止对这个 Worker 的任务调度，把失效 Worker 的任务调度到其他 Worker 上重新执行。

3. 案例分析

单词计数(Word Count)是一个经典的问题，也是能体现 MapReduce 设计思想的最简单算法之一。该算法主要是为了完成对文字数据中所出现的单词进行计数，如图 10-4 所示。

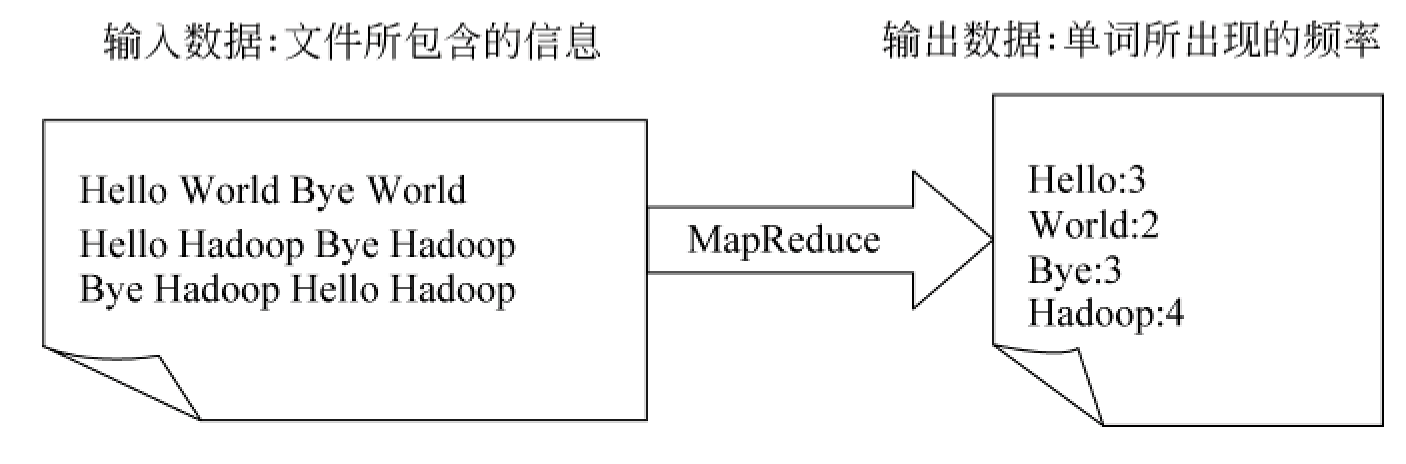


图 10-4 单词计数示意图

伪代码如下：

```
Map(K,V){
    For each word w in V
    Collect(w, 1);
}
Reduce(K,V[ ]){
    int count=0;
    For each v in V
    count+=v;
    Collect(K, count);
}
```

下面就根据 MapReduce 的四个执行步骤对这一算法进行详细的介绍。

根据文件所包含的信息分割(Split)文件，在这里把文件的每行分割为一组，共三组，如图 10-5 所示。这一步由系统自动完成。

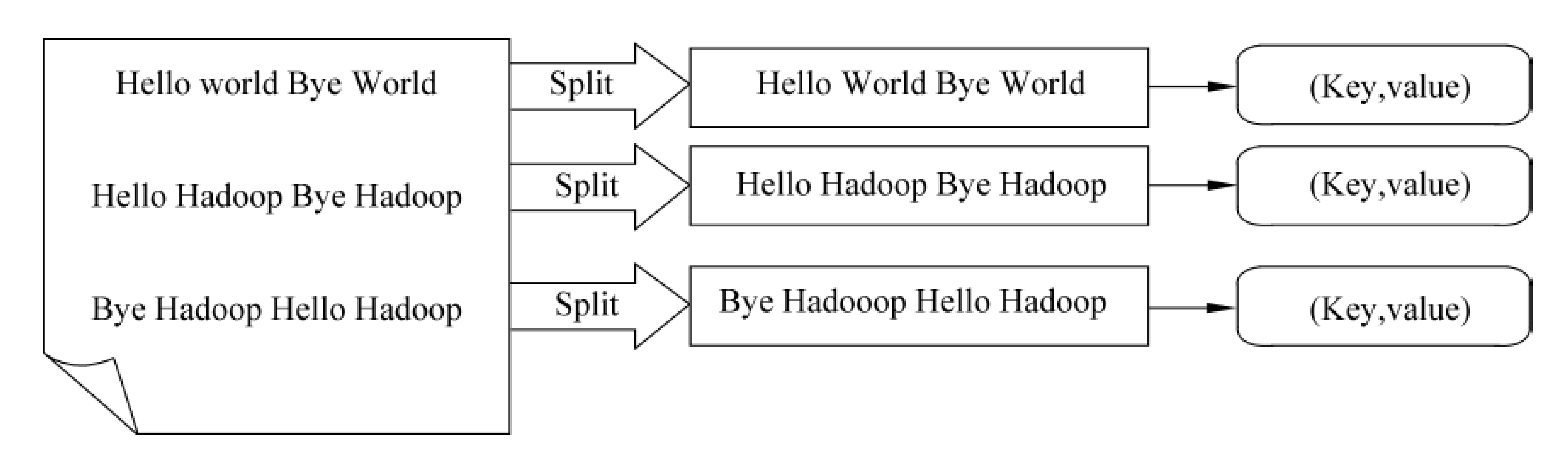


图 10-5 分割过程示意图

对分割之后的每一对<key,value>利用用户定义的 Map 进行处理，再生成新的<key,value>对，如图 10-6 所示。

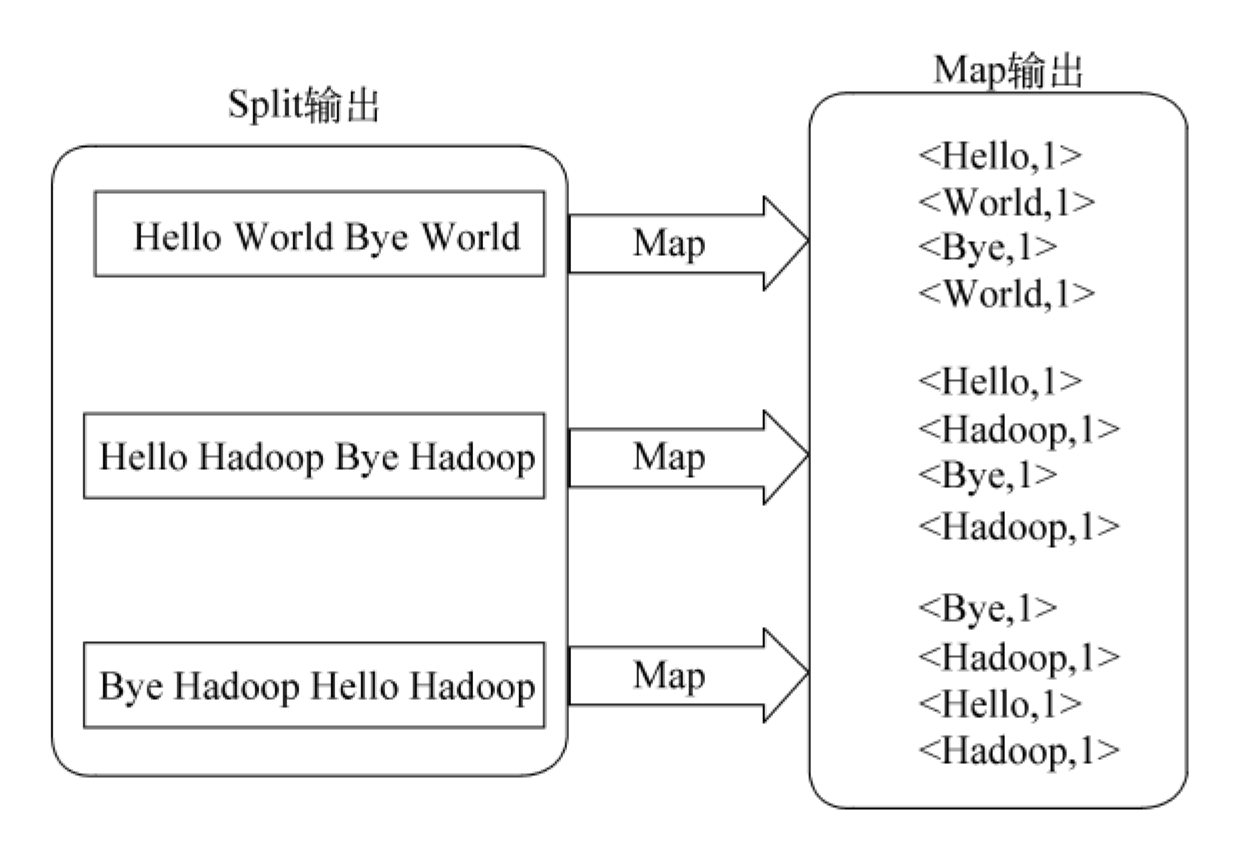


图 10-6　Map 过程示意图

Map 输出之后有一个内部的 Fold 过程，与第一步一样，都是由系统自动完成的，如图 10-7 所示。

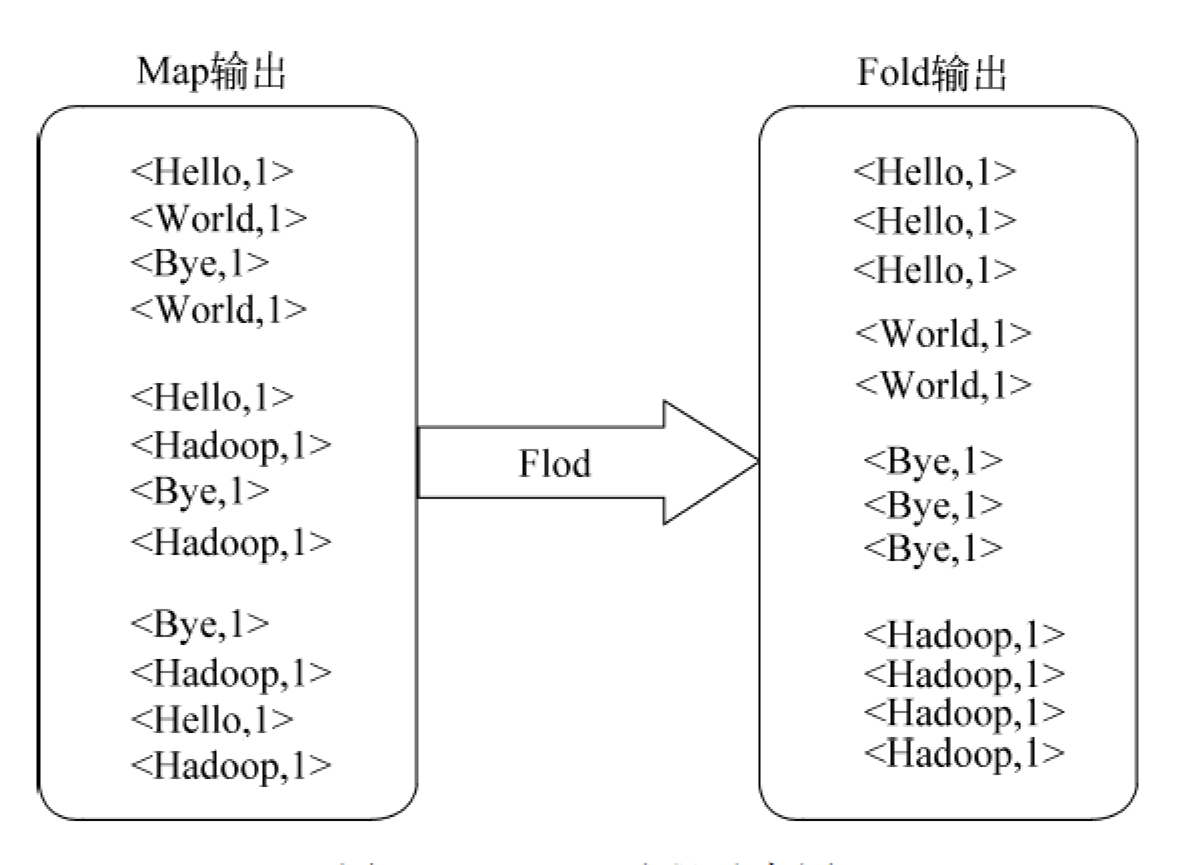


图 10-7　Fold 过程示意图

经过 Fold 步骤之后的输出与结果已经非常接近，再由用户定义的 Reduce 步骤完成最后的工作即可，如图 10-8 所示。

10.4.7　分布式锁服务 Chubby

Chubby 是 Google 设计的提供粗粒度锁服务的一个文件系统，它基于松耦合分布式系统，解决了分布的一致性问题。通过使用 Chubby 的锁服务，用户可以确保数据操作过程中的一致性。不过值得注意的是，这种锁只是一种建议性的锁（Advisory Lock）而不是强制性的锁（Mandatory Lock），如此选择的目的是使系统具有更大的灵活性。

GFS 使用 Chubby 来选取一个 GFS 主服务器，Bigtable 使用 Chubby 指定一个主服务器并发现、控制与其相关的子表服务器。除了最常用的锁服务之外，Chubby 还可以作为一个稳定的存储系统存储包括元数据在类的小数据。同时 Google 内部还使用 Chubby 进行名字服务（Name Server）。本节首先简要介绍 Paxos 算法，因为 Chubby 内部一致性问题的实现用到了 Paxos 算法；然后围绕 Chubby 系统的设计和实现展开讲解。通过本节的学习，

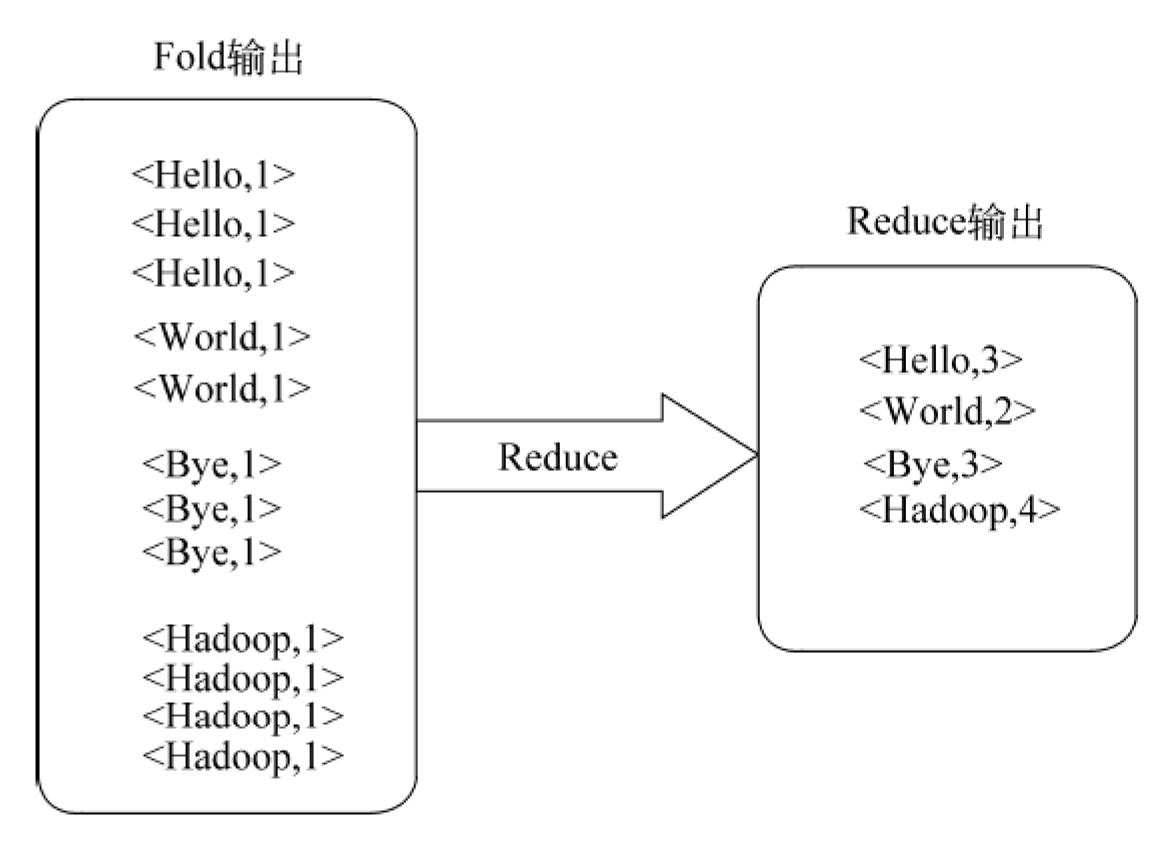


图 10-8 Reduce 过程示意图

读者应该对分布式系统中一致性问题的一般性算法有了初步的了解，着重掌握 Chubby 系统设计和实现的精髓。

1. Paxos 算法

Paxos 算法是由供职于微软的 Leslie Lamport 最先提出的一种基于消息传递(Messages Passing)的一致性算法。在目前所有的一致性算法中，该算法最常用且被认为是最有效的。要想了解 Paxos 算法，我们首先需要知道什么是分布式系统中的一致性问题，因为 Paxos 算法就是为了解决这个问题而提出的。简单地说分布式系统的一致性问题，就是如何保证系统中初始状态相同的各个节点在执行相同的操作序列时，看到的指令序列是完全一致的，并且最终得到完全一致的结果。在 Lamport 提出的 Paxos 算法中节点被分成了三种类型：proposers、acceptors 和 learners。其中 proposers 提出决议(Value)，acceptors 批准决议，learners 获取并使用已经通过的决议。一个节点可以兼有多重类型。在这种情况下，满足以下三个条件就可以保证数据的一致性。

(1) 决议只有在被 proposers 提出后才能批准。

(2) 每次只批准一个决议。

(3) 只有决议确定被批准后 learners 才能获取这个决议。

Lamport 通过约束条件的不断加强，最后得到了一个可以实际运用到算法中的完整约束条件：如果一个编号为 n 的提案具有值 v，那么存在一个多数派，要么他们中没有人批准过编号小于 n 的任何提案，要么他们进行的最近一次批准具有值 v。为了保证决议的唯一性，acceptors 也要满足一个如下的约束条件：当且仅当 acceptors 没有收到编号大于 n 的请求时，acceptors 才批准编号为 n 的提案。

在这些约束条件的基础上，可以将一个决议的通过分成两个阶段。

准备阶段：proposers 选择一个提案并将它的编号设为 n，然后将它发送给 acceptors 中的一个多数派。Acceptors 收到后，如果提案的编号大于它已经回复的所有消息，则 acceptors 将自己上次的批准回复给 proposers，并不再批准小于 n 的提案。

批准阶段：当 proposers 接收到 acceptors 中的这个多数派的回复后，就向回复请求的 acceptors 发送 accept 请求，在符合 acceptors 一方的约束条件下，acceptors 收到 accept 请求后即批准这个请求。

为了减少决议发布过程中的消息量，acceptors 将这个通过的决议发送给 learners 的一个子集，然后由这个子集中的 learners 去通知所有其他的 learners。一般情况下，以上的算法过程就可以成功地解决一致性问题，但是也有特殊情况。根据算法，一个编号更大的提案会终止之前的提案过程，如果两个 proposer 在这种情况下都转而提出一个编号更大的提案，那么就可能陷入活锁。此时需要选举出一个 president，仅允许 president 提出提案。

以上只是简要地向大家介绍了 Paxos 算法的核心内容，更多的实现细节读者可以参考 Lamport 关于 Paxos 算法实现的文章。

2. Chubby 系统设计

通常情况下 Google 的一个数据中心仅运行一个 Chubby 单元（Chubby cell，下面会有详细讲解），而这个单元需要支持包括 GFS、Bigtable 在内的众多 Google 服务。这种苛刻的服务要求使得 Chubby 在设计之初就要充分考虑到系统需要实现的目标以及可能出现的各种问题。

Chubby 的设计目标主要有以下几点。

（1）高可用性和高可靠性。这是系统设计的首要目标，在保证这一目标的基础上再考虑系统的吞吐量和存储能力。

（2）高扩展性。将数据存储在价格较为低廉的 RAM，支持大规模用户访问文件。

（3）支持粗粒度的建议性锁服务。提供这种服务的根本目的是提高系统的性能。

（4）服务信息的直接存储。可以直接存储包括元数据、系统参数在内的有关服务信息，而不需要再维护另一个服务。

（5）支持通报机制。客户可以及时地了解到事件的发生情况。

（6）支持缓存机制。通过一致性缓存将常用信息保存在客户端，避免了频繁地访问主服务器。

前面提到在分布式系统中保持数据一致性最常用也最有效的算法是 Paxos，很多系统就是将 Paxos 算法作为其一致性算法的核心。但是 Google 并没有直接实现一个包含了 Paxos 算法的函数库，相反，Google 设计了一个全新的锁服务 Chubby。Google 做出这种设计主要是考虑到以下几个问题。

（1）通常情况下开发者在开发的初期很少考虑系统的一致性问题，但是随着开发的不断进行，这种问题会变得越来越严重。单独的锁服务可以保证原有系统的架构不会发生改变，而使用函数库很可能需要对系统的架构做出大幅度的改动。

（2）系统中很多事件的发生是需要告知其他用户和服务器的，使用一个基于文件系统的锁服务可以将这些变动写入文件中。这样其他需要了解这些变动的用户和服务器直接访问这些文件即可，避免了因大量的系统组件之间的事件通信带来的系统性能下降。

（3）基于锁的开发接口容易被开发者接受。虽然在分布式系统中锁的使用会有很大的不同，但是和一致性算法相比，锁显然被更多的开发者所熟知。

一般来说分布式一致性问题通过 quorum 机制（简单来说就是根据少数服从多数的选举原则产生一个决议）做出决策，为了保证系统的高可用性，需要若干台机器，但是使用单独的锁服务，一台机器也能保证这种高可用性。也就是说，Chubby 在自身服务的实现时利用若干台机器实现了高可用性，而外部用户利用 Chubby 则只需一台机器就可以保证高可用性。

正是考虑到以上几个问题，Google设计了Chubby，而不是单独地维护一个函数库（实际上，Google有这样一个独立于Chubby的函数库，不过一般情况下并不会使用）。在设计的过程中有一些细节问题也值得我们关注，比如在Chubby系统中采用了建议性的锁而没有采用强制性的锁。两者的根本区别在于用户访问某个被锁定的文件时，建议性的锁不会阻止这种行为，而强制性的锁则会阻止这种行为，实际上这是为了便于系统组件之间的信息交互行为。另外Chubby还采用了粗粒度（Coarse-Grained）锁服务而没有采用细粒度（Fine-Grained）锁服务，两者的差异在于持有锁的时间。细粒度的锁持有时间很短，常常只有几秒甚至更少，而粗粒度的锁持有的时间可长达几天，做出如此选择的目的是减少频繁换锁带来的系统开销。当然用户也可以自行实现细粒度锁，不过建议还是使用粗粒度的锁。

图10-9就是Chubby的基本架构。很明显，Chubby被划分成两个部分：客户端和服务器端，客户端和服务器端之间通过远程过程调用（RPC）来连接。在客户这一端每个客户应用程序都有一个Chubby程序库（Chubby Library），客户端的所有应用都是通过调用这个库中的相关函数来完成的。服务器一端称为Chubby单元，一般是由五个称为副本（Replica）的服务器组成的，这五个副本在配置上完全一致，并且在系统刚开始时处于对等地位。这些副本通过quorum机制选举产生一个主服务器（Master），并保证在一定的时间内有且仅有一个主服务器，这个时间就称为主服务器租约期（Master Lease）。如果某个服务器被连续推举为主服务器，这个租约期就会不断地被更新。租续期内所有的客户请求都是由主服务器来处理的。客户端如果需要确定主服务器的位置，可以向DNS发送一个主服务器定位请求，非主服务器的副本将对该请求做出回应，通过这种方式客户端能够快速、准确地对主服务器做出定位。

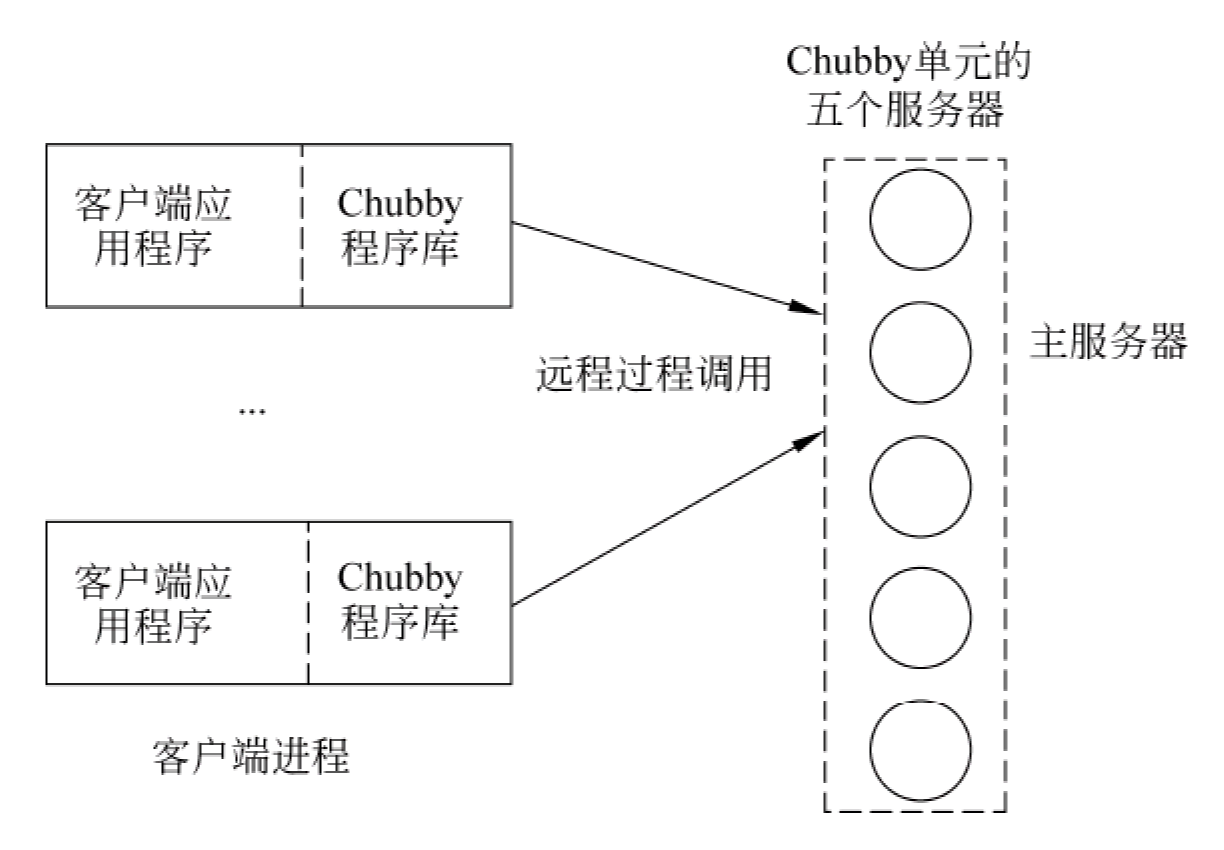


图10-9　Chubby的基本架构示意图

3. Chubby文件系统

Chubby系统本质上就是一个分布式的、存储大量小文件的文件系统，它所有的操作都是在文件的基础上完成的。例如在Chubby最常用的锁服务中，每一个文件就代表了一个锁，用户通过打开、关闭和读取文件，获取共享（Shared）锁或独占（Exclusive）锁。选举主服务器的过程中，符合条件的服务器都同时申请打开某个文件并请求锁住该文件。成功获得锁的服务器自动成为主服务器并将其地址写入这个文件夹，以便其他服务器和用户可以获

知主服务器的地址信息。

Chubby 的文件系统和 UNIX 类似。例如在文件名"/ls/foo/wombat/pouch"中，ls 代表 lock service，这是所有 Chubby 文件系统的共有前缀；foo 是某个单元的名称；/wombat/pouch 则是 foo 这个单元上的文件目录或者文件名。由于 Chubby 自身的特殊服务要求，Google 对 Chubby 做了一些与 UNIX 不同的改变。例如 Chubby 不支持内部文件的移动，不记录文件的最后访问时间，另外在 Chubby 中并没有符号连接(Symbolic Link，又叫软连接，类似于 Windows 系统中的快捷方式)和硬连接(Hard Link，类似于别名)的概念。在具体实现时，文件系统由许多节点组成，分为永久型和临时型，每个节点就是一个文件或目录。节点中保存着包括 ACL(Access Control List，访问控制列表)在内的多种系统元数据。为了用户能够及时了解元数据的变动，系统规定每个节点的元数据都应当包含以下四种单调递增的 64 位编号。

- 实例号(Instance Number)：新节点实例号必定大于旧节点的实例号。
- 内容生成号(Content Generation Number)：文件内容修改时该号增加。
- 锁生成号(Lock Generation Number)：锁被用户持有时该号增加。
- ACL 生成号(ACL Generation Number)：ACL 名被覆写时该号增加。

用户在打开某个节点时就会获取一个类似于 UNIX 中文件描述符(File Descriptor)的句柄(Handles)，这个句柄由以下三个部分组成。

- 校验数位(Check Digit)：防止其他用户创建或猜测这个句柄。
- 序号(Sequence Number)：用来确定句柄是由当前还是以前的主服务器创建的。
- 模式信息(Mode Information)：用于新的主服务器重新创建一个旧的句柄。

在实际的执行中，为了避免所有的通信都使用序号带来的系统开销增长，Chubby 引入了 sequencer 的概念。sequencer 实际上就是一个序号，只不过这个序号只能由锁的持有者在获取锁时向系统发出请求来获得。这样一来 Chubby 系统中只有涉及锁的操作才需要序号，其他一概不用。在文件操作中，用户可以将句柄看作一个指向文件系统的指针。这个指针支持一系列的操作，常用的句柄操作函数如表 10-1 所示。

表 10-1　常用句柄函数及其作用

函数名称	作　　用
Open()	打开某个文件或者目录来创建句柄
Close()	关闭打开的句柄，后续的任何操作都将中止
Poison()	中止当前未完成及后续的操作，但不关闭句柄
GetContentsAndStat()	返回文件内容及元数据
GetStat()	只返回文件元数据
ReadDir()	返回子目录名称及其元数据
SetContents()	向文件中写入内容
SetACL()	设置 ACL 名称
Delete()	如果该节点没有子节点，则执行删除操作

续表

函数名称	作　用
Acquire()	获取锁
Release()	释放锁
GetSequencer()	返回一个 sequencer
SetSequencer()	将 sequencer 与某个句柄进行关联
CheckSequencer()	检查某个 sequencer 是否有效

10.4.8 通信协议

客户端和主服务器之间的通信是通过 KeepAlive 握手协议来维持的，图 10-10 就是这一通信过程的简单示意图。

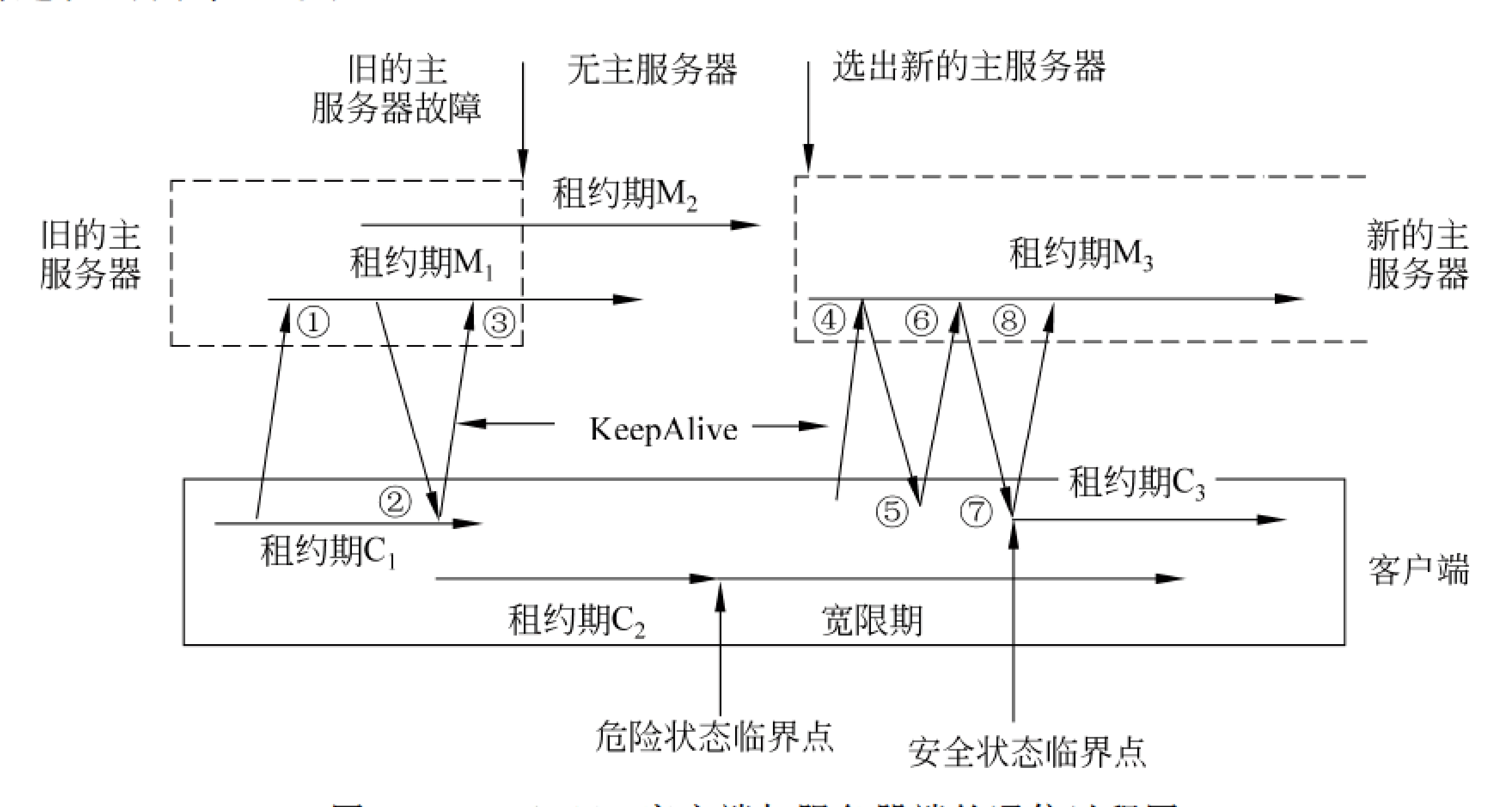


图 10-10　Chubby 客户端与服务器端的通信过程图

图 10-11 中从左到右时间在增加，斜向上的箭头表示一次 KeepAlive 请求，斜向下的箭头则是主服务器的一次回应。M_1、M_2、M_3 表示不同的主服务器租约期。C_1、C_2、C_3 则是客户端对主服务器租约期时长做出的一个估计。KeepAlive 是周期发送的一种信息，它主要有两方面的功能：延迟租约的有效期和携带事件信息告诉用户更新。主要的事件包括文件内容被修改、子节点的增加、删除和修改、主服务器出错、句柄失效等。正常情况下，通过 KeepAlive 握手协议租约期会得到延长，事件也会及时地通知给用户。但是由于系统有一定的失效概率，引入故障处理措施是很有必要的。通常情况下系统可能会出现两种故障：客户端租约期过期和主服务器故障，对于这两种情况系统有着不同的应对方式。

1. 客户端租约过期

刚开始时，客户端向主服务器发出一个 KeepAlive 请求（图 10-10 中的①），如果有需要通知的事件时，则主服务器会立刻做出回应，否则主服务器并不立刻对这个请求做出回应，而是等到客户端的租约期 C_1 快结束的时候才做出回应（图 10-10 中的②），并更新主服务器租约期为 M_2。客户端在接到这个回应后认为该主服务器仍处于活跃状态，于是将租约期

更新为 C_2 并立刻发出新的 KeepAlive 请求(图 10-10 中的③)。同样的,主服务器可能不是立刻回应而是等待 C_2 接近结束,但是在这个过程中主服务器出现故障停止使用。在等待了一段时间后 C_2 到期,由于并没有收到主服务器的回应,系统向客户端发出一个危险(Jeopardy)事件,客户端清空并暂时停用自己的缓存,从而进入一个称为宽限期(Grace Period)的危险状态。这个宽限期默认是 45 秒。在宽限期内,客户端不会立刻断开其与服务器端的联系,而是不断地做探询。图 10-10 中新的主服务器很快被重新选出,当它接到客户端的第一个 KeepAlive 请求(图 10-10 中的④)时会拒绝(图 10-10 中的⑤),因为这个请求的纪元号(Epoch Number)错误。不同主服务器的纪元号不相同,客户端的每次请求都需要这个号来保证处理的请求是针对当前的主服务器。客户端在主服务器拒绝之后会使用新的纪元号来发送 KeepAlive 请求(图 10-10 中的⑥)。新的主服务器接受这个请求并立刻做出回应(图 10-10 中的⑦)。如果客户端接收到这个回应的时间仍处于宽限期内,则系统会恢复到安全状态,租约期更新为 C_3。如果在宽限期未接到主服务器的相关回应,则客户端终止当前的会话。

2. 主服务器出错

在客户端和主服务器端进行通信时可能会遇到主服务器故障。正常情况下旧的主服务器出现故障后系统会很快地选举出新的主服务器,新选举的主服务器在完全运行前需要经历以下 9 个步骤。

① 产生一个新的纪元号以便今后客户端通信时使用,这能保证当前的主服务器不必处理针对旧的主服务器的请求。

② 只处理主服务器位置相关的信息,不处理会话相关的信息。

③ 构建处理会话和锁所需的内部数据结构。

④ 允许客户端发送 KeepAlive 请求,不处理其他会话相关的信息。

⑤ 向每个会话发送一个故障事件,促使所有的客户端清空缓存。

⑥ 等待直到所有的会话都收到故障事件或会话终止。

⑦ 开始允许执行所有的操作。

⑧ 如果客户端使用了旧的句柄,则需要为其重新构建新的句柄。

⑨ 一定时间段后(1 分钟),删除没有被打开过的临时文件夹。

如果这一过程在宽限期内顺利完成,则用户不会感觉到任何故障的发生,也就是说新旧主服务器的替换对于用户来说是透明的,用户感觉到的仅仅是一个延迟。使用宽限期的好处正是如此。

在系统实现时,Chubby 还使用了一致性客户端缓存(Consistent Client-Side Caching)技术,这样做的目的是减少通信压力,降低通信频率。在客户端保存一个和单元上数据一致的本地缓存,这样需要时客户可以直接从缓存中取出数据而不用再和主服务器通信。当某个文件数据或者元数据需要修改时,主服务器首先将这个修改阻塞;然后通过查询主服务器自身维护的一个缓存表,向所有对修改的数据进行了缓存的客户端发送一个无效标志(Invalidation);客户端收到这个无效标志后会返回一个确认(Acknowledge),主服务器在收到所有的确认后才解除阻塞并完成这次修改。这个过程的执行效率非常高,仅仅需要发送一次无效标志即可,因为主服务器对于没有返回确认的节点就直接认为其是未缓存的。

10.4.9　正确性与性能

1. 一致性

前面提到过每个 Chubby 单元是由五个副本组成的，这五个副本中需要选举产生一个主服务器，这种选举本质上就是一个一致性问题。在实际的执行过程中，Chubby 使用 Paxos 算法来解决这个问题。

主服务器产生后客户端的所有读写操作都是由主服务器来完成的。读操作很简单，客户直接从主服务器上读取所需数据即可，但是写操作就涉及数据一致性的问题了。为了保证客户的写操作能够同步到所有的服务器上，系统再次利用了 Paxos 算法。因此，可以看出 Paxos 算法在分布式一致性问题中的作用是巨大的。

2. 安全性

Chubby 采用的是 ACL 形式的安全保障措施。系统中有三种 ACL 名，分别是写 ACL 名(Write ACL Name)、读 ACL 名(Read ACL Name)和变更 ACL 名(Change ACL Name)。只要不被覆写，子节点都是直接继承父节点的 ACL 名。ACL 同样被保存在文件中，它是节点元数据的一部分，用户在进行相关操作时首先需要通过 ACL 来获取相应的授权。图 10-11 是一个用户成功写文件所需经历的过程。

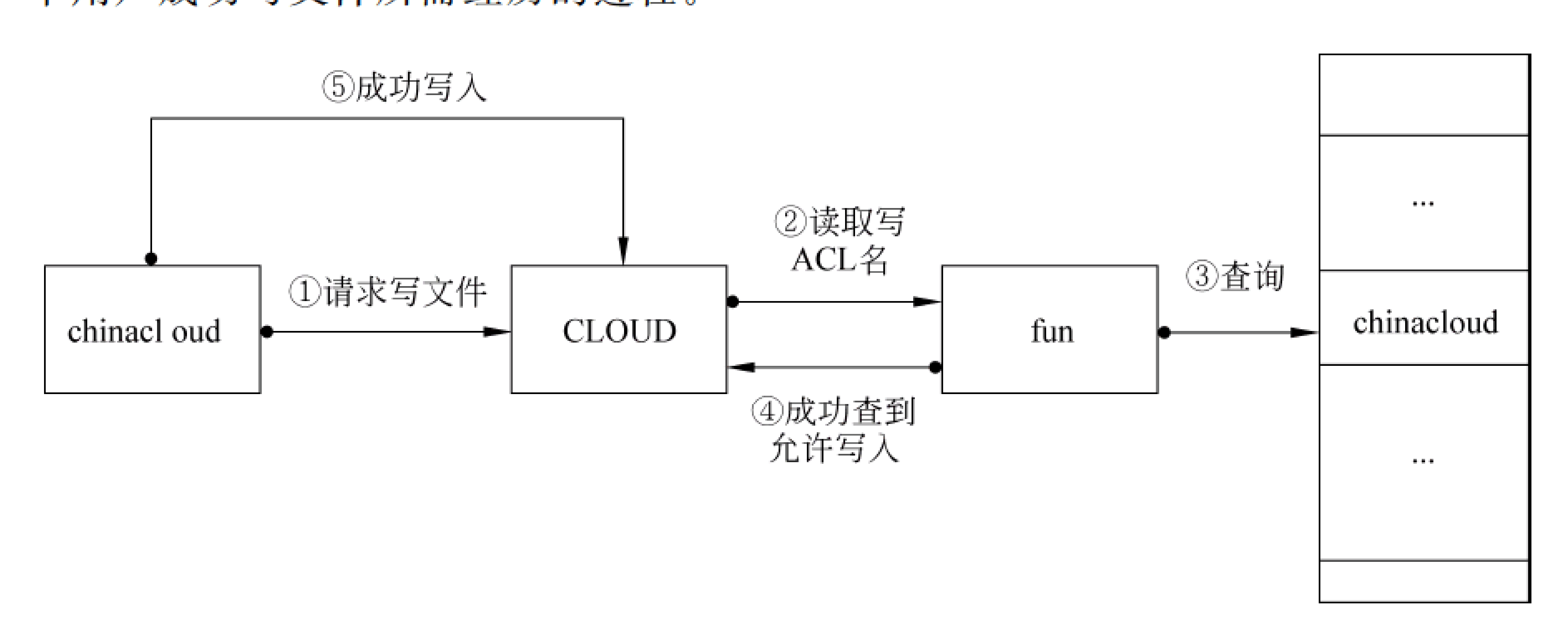


图 10-11　Chubby 的 ACL 机制

用户 chinacloud 请求向文件 CLOUD 中写入内容。CLOUD 首先读取自身的写 ACL 名是 fun，接着在 fun 中查到了 chinacloud 这一行记录，于是返回信息允许 chinacloud 对文件进行写操作，此时 chinacloud 才被允许向 CLOUD 写入内容。其他的操作和写操作类似。

3. 性能优化

为了满足系统的高可扩展性，Chubby 目前已经采取了一些措施。比如提高主服务器默认的租约期、使用协议转换服务将 Chubby 协议转换成较简单的协议。还有就是使用上面提到的客户端一致性缓存。除此之外，Google 的工程师们还考虑使用代理(Proxy)和分区(Partition)技术，虽然目前这两种技术并没有实际使用，但是在设计的时候还是被包含进系统，不排除将来使用的可能。代理可以减少主服务器处理 KeepAlive 以及读请求带来的服务器负载，但是它并不能减少写操作带来的通信量。不过根据 Google 自己的数据统计表明，在所有的请求中，写请求仅占极少的一部分，几乎可以忽略不计。使用分区技术可以将一个单元的命名空间(Name Space)划分成 N 份。除了少量的跨分区通信外，大部分的分区

都可以独自地处理服务请求。通过分区可以减少各个分区上的读写通信量,但不能减少KeepAlive请求的通信量。因此,如果需要,将代理和分区技术结合起来使用才可以明显提高系统同时处理的服务请求量。

10.4.10　分布式结构化数据表 Bigtable

Bigtable是 Google开发的基于 GFS和 Chubby的分布式存储系统。Google的很多数据,包括 Web索引、卫星图像数据等在内的海量结构化和半结构化数据,都是存储在Bigtable中的。从实现上来看,Bigtable并没有什么全新的技术,但是如何选择合适的技术并将这些技术高效、巧妙地结合在一起,恰恰是最大的难点。Google的工程师通过研究以及大量的实践,完美实现了相关技术的选择及融合。Bigtable在很多方面和数据库类似,但它并不是真正意义上的数据库。通过本节的学习,读者将会对 Bigtable的数据模型、系统架构、实现以及使用的一些数据库技术有一个全面的认识。

1. 动机与目标

(1) 动机

Google设计 Bigtable的动机主要有如下三个方面。

① 需要存储的数据种类繁多。Google目前向公众开放的服务很多,需要处理的数据类型也非常多,包括 URL、网页内容、用户的个性化设置在内的数据都是 Google需要经常处理的。

② 海量的服务请求。Google运行着目前世界上最繁忙的系统,它每时每刻处理的客户服务请求数量是普通的系统根本无法承受的。

③ 商用数据库无法满足 Google的需求。一方面现有商用数据库的设计着眼点在于其通用性,面对 Google的苛刻服务要求根本无法满足,而且在数量庞大的服务器上根本无法成功部署普通的商用数据库。另一方面对于底层系统的完全掌控会给后期的系统维护、升级带来极大的便利。

(2) 目标

在仔细考察了 Google的日常需求后,Bigtable开发团队确定了 Bigtable设计所需达到的如下几个基本目标。

① 广泛的适用性。Bigtable是为了满足一系列 Google产品而并非特定产品的存储要求。

② 很强的可扩展性。根据需要随时可以加入或撤销服务器。

③ 高可用性。对于客户来说,有时候即使短暂的服务中断也是不能忍受的。Bigtable设计的重要目标之一就是确保几乎所有的情况下系统都可用。

④ 简单性。底层系统的简单性既可以减少系统出错的概率,也为上层应用的开发带来便利。

在目标确定之后,Google开发者就在现有的数据库技术中进行了大规模的筛选,希望各种技术之间能够扬长避短,巧妙地结合起来。最终实现的系统也确实达到了原定的目标。下面就开始详细讲解 Bigtable。

2. 数据模型

Bigtable是一个分布式多维映射表,表中的数据是通过一个行关键字(Row Key)、一个

列关键字(Column Key)以及一个时间戳(Time Stamp)进行索引的。Bigtable 对存储在其中的数据不做任何解析,一律看作字符串,具体数据结构的实现需要用户自行处理。Bigtable 的存储逻辑可以表示为:(row: string, column: string, time: int64)→string。Bigtable 数据的存储格式如图 10-12 所示。

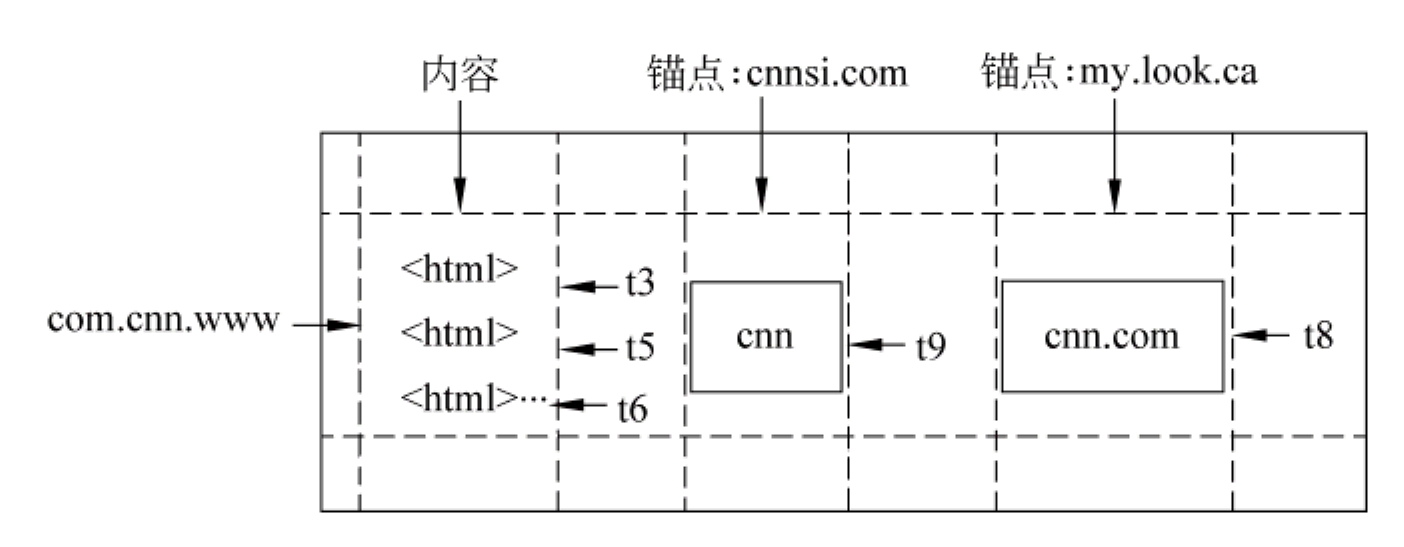


图 10-12　Bigtable 数据模型

(1) 行

Bigtable 的行关键字可以是任意的字符串,但是大小不能够超过 64KB。Bigtable 和传统的关系型数据库有很大不同,它不支持一般意义上的事务,但能保证对于行的读写操作具有原子性(Atomic)。表中数据都是根据行关键字进行排序的,排序使用的是词典序。图 10-12 是 Bigtable 数据模型的一个典型实例,其中 com. cnn. www 就是一个行关键字。不直接存储网页地址而将其倒排,这是 Bigtable 的一个巧妙设计。这样做至少会带来以下两个好处。

① 同一地址域的网页会被存储在表中的连续位置,有利于用户查找和分析。

② 倒排便于数据压缩,可以大幅提高压缩率。

单个的大表由于规模问题不利于数据的处理,因此 Bigtable 将一个表分成了很多子表(Tablet),每个子表包含多个行。子表是 Bigtable 中数据划分和负载均衡的基本单位。有关子表的内容会在稍后详细讲解。

(2) 列

Bigtable 并不是简单地存储所有的列关键字,而是将其组织成所谓的列族(Column Family),每个族中的数据都属于同一个类型,并且同族的数据会被压缩在一起保存。引入了列族的概念之后,列关键字就采用下述的语法规则来定义。

族名: 限定词(family: qualifier)

族名必须有意义,限定词则可以任意选定。在图 10-12 中内容(Contents)、锚点(Anchor,就是 HTML 中的链接)都是不同的族。而 cnnsi. com 和 my. look. ca 则是锚点族中不同的限定词。通过这种方式组织的数据结构清晰明了,含义也很清楚。族同时也是 Bigtable 中访问控制(Access Control)的基本单元,也就是说访问权限的设置是在族这一级别上进行的。

(3) 时间戳

Google 的很多服务比如网页检索和用户的个性化设置等都需要保存不同时间的数据,这些不同的数据版本必须通过时间戳来区分。图 10-12 中内容列的 t3、t5 和 t6 表明其中保列的 t3、t5 和 t6 表明其中保存了在 t3、t5 和 t6 这三个时间获取的网页。Bigtable 中的时间戳是 64 位整型数,具体的赋值方式可以采取系统默认的方式,也可以由用户自行定义。

为了简化不同版本的数据管理，Bigtable 目前提供了两种设置：一种是保留最近的 N 个不同版本，图 10-12 中数据模型采取的就是这种方法，它保存最新的三个版本数据。另一种就是保留限定时间内的所有不同版本，比如可以保存最近 10 天的所有不同版本数据。失效的版本将会由 Bigtable 的垃圾回收机制自动处理。

3. 系统架构

Bigtable 是在 Google 的另外三个云计算组件基础之上构建的，其基本架构如图 10-13 所示。

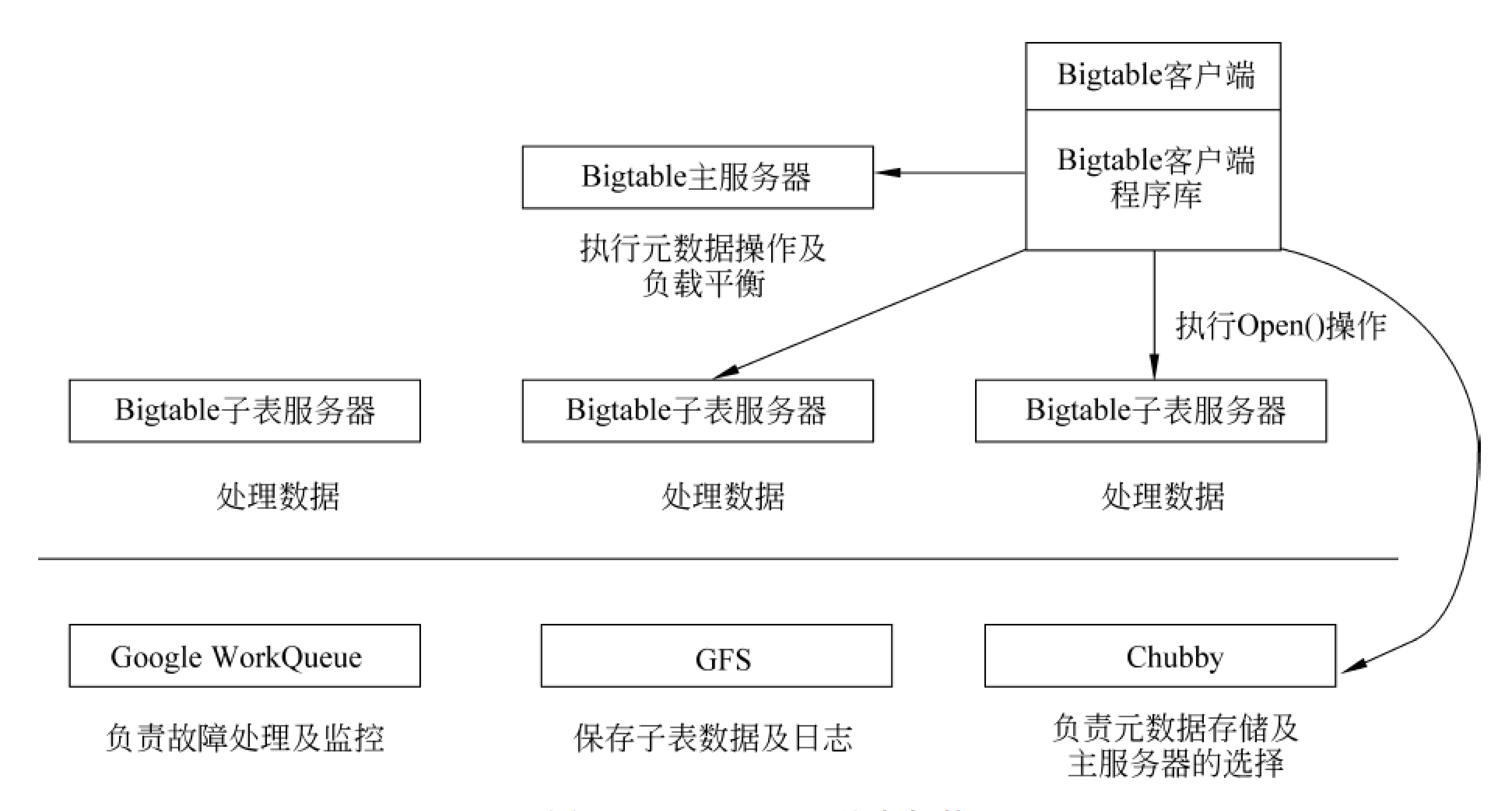


图 10-13　Bigtable 基本架构

图中 WorkQueue 是一个分布式的任务调度器，它主要被用来处理分布式系统队列分组和任务调度，关于其实现，Google 并没有公开。在前面已经讲过，GFS 是 Google 的分布式文件系统，在 Bigtable 中 GFS 主要用来存储子表数据以及一些日志文件。Bigtable 还需要一个锁服务的支持，Bigtable 选用了 Google 自己开发的分布式锁服务 Chubby。在 Bigtable 中 Chubby 主要有以下几个作用。

① 选取并保证同一时间内只有一个主服务器(Master Server)。

② 获取子表的位置信息。

③ 保存 Bigtable 的模式信息及访问控制列表。

另外在 Bigtable 的实际执行过程中，Google 的 MapReduce 和 Sawzall 也被用来改善其性能，不过需要注意的是这两个组件并不是实现 Bigtable 所必需的。

Bigtable 主要由三个部分组成：客户端程序库(Client Library)、一个主服务器(Master Server)和多个子表服务器(Tablet Server)，这三个部分在图 10-13 中都有相应的表示。从中可以看出，客户需要访问 Bigtable 服务时首先要利用其库函数执行 Open 操作来打开一个锁(实际上就是获取了文件目录)，锁打开以后客户端就可以和子表服务器进行通信了。与许多具有单个主节点的分布式系统一样，客户端主要与子表服务器通信，几乎不和主服务器进行通信，这使得主服务器的负载大大降低。主服务主要进行一些元数据的操作以及子表服务器之间的负载调度问题，实际的数据是存储在子表服务器上的。客户程序库的概念

比较简单,这里不做讲解,下面对主服务器和子表服务器展开讲解。

4. 主服务器

主服务的主要作用如图 10-14 所示。

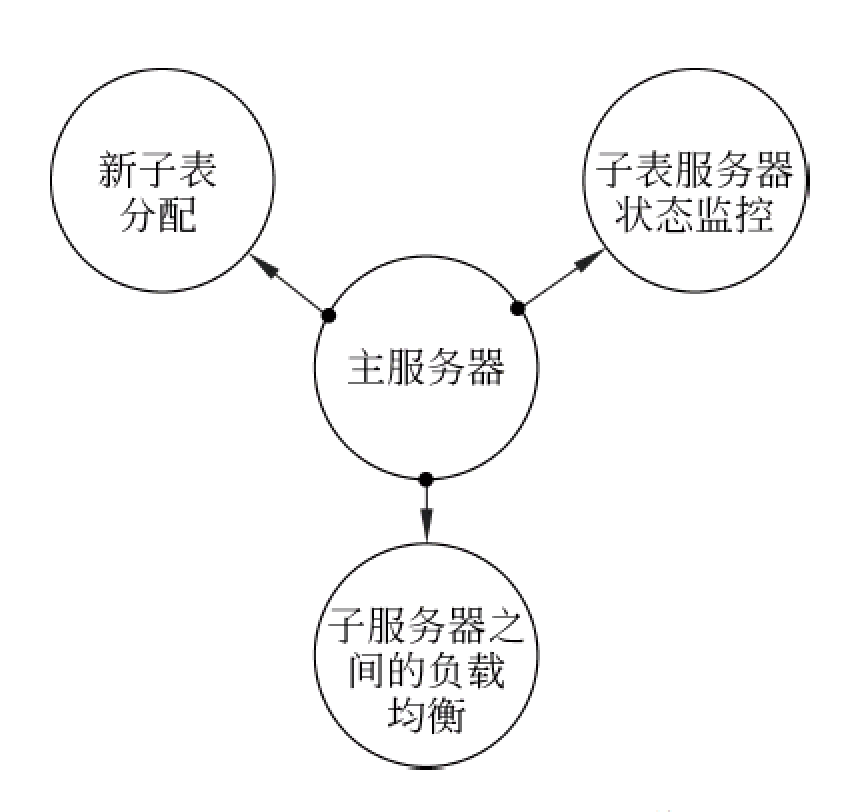


图 10-14 主服务器的主要作用

当一个新的子表产生时,主服务器通过一个加载命令将其分配给一个空间足够的子表服务器。创建新表、表合并以及较大子表的分裂都会产生一个或多个新子表。对于前面两种,主服务器会自动检测到,因为这两个操作是由主服务器发起的,而较大子表的分裂是由子服务发起并完成的,所以主服务器并不能自动检测到,因此在分割完成之后子服务器需要向主服务发出一个通知。由于系统设计之初就要求能达到良好的扩展性,所以主服务器必须对子表服务器的状态进行监控,以便及时检测到服务器的加入或撤销。Bigtable 中主服务器对子表服务器的监控是通过 Chubby 来完成的,子表服务器在初始化时都会从 Chubby 中得到一个独占锁。通过这种方式,所有的子表服务器基本信息被保存在 Chubby 中一个称为服务器目录(Server Directory)的特殊目录之中。主服务器通过检测这个目录就可以随时获取最新的子表服务器信息,包括目前活跃的子表服务器,以及每个子表服务器上已经分配的子表。对于每个具体的子表服务器,主服务器会定期向其询问独占锁的状态。如果子表服务器的锁丢失或没有回应,则此时可能有两种情况,要么是 Chubby 出现了问题(虽然这种概率很小,但的确存在,Google 自己也做过相关测试),要么是子表服务器自身出现了问题。对此主服务器首先自己尝试获取这个独占锁,如果失败说明 Chubby 服务出现问题,需等待 Chubby 服务的恢复。如果成功则说明 Chubby 服务良好而子表服务器本身出现了问题。这种情况下主服务器会中止这个子表服务器并将其上的子表全部移至其他子表服务器。当在状态监测时发现某个子表服务器上负载过重时,主服务器会自动对其进行负载均衡操作。

基于系统出现故障是一种常态的设计理念(Google 几乎所有的产品都是基于这个设计理念),每个主服务器被设定了一个会话时间的限制。当某个主服务器到时退出后,管理系统就会指定一个新的主服务器,这个主服务器的启动需要经历以下四个步骤。

① 从 Chubby 中获取一个独占锁,确保同一时间只有一个主服务器。

② 扫描服务器目录,发现目前活跃的子表服务器。

③ 与所有的活跃子表服务器取得联系以便了解所有子表的分配情况。

④ 通过扫描元数据表(Metadata Table),发现未分配的子表并将其分配到合适的子表服务器。如果元数据表未分配,则首先需要将根子表(Root Tablet)加入未分配的子表中。由于根子表保存了其他所有元数据子表的信息,确保了扫描能够发现所有未分配的子表。

在成功完成以上四个步骤后,主服务器就可以正常运行了。

5. 表服务器

Bigtable 中实际的数据都是以子表的形式保存在子表服务器上的,客户一般也只和子表服务器进行通信,所以子表以及子表服务器是我们重点讲解的概念。子表服务器上的操作主要涉及子表的定位、分配以及子表数据的最终存储问题。其中子表分配在前面已经有

了详细介绍，这里略过不讲。在讲解其他问题之前我们首先介绍一下 SSTable 的概念以及子表的基本结构。

(1) SSTable 及子表基本结构

SSTable 是 Google 为 Bigtable 设计的内部数据存储格式。所有的 SSTable 文件都是存储在 GFS 上的，用户可以查询相应的值，图 10-15 是 SSTable 格式的基本示意图。

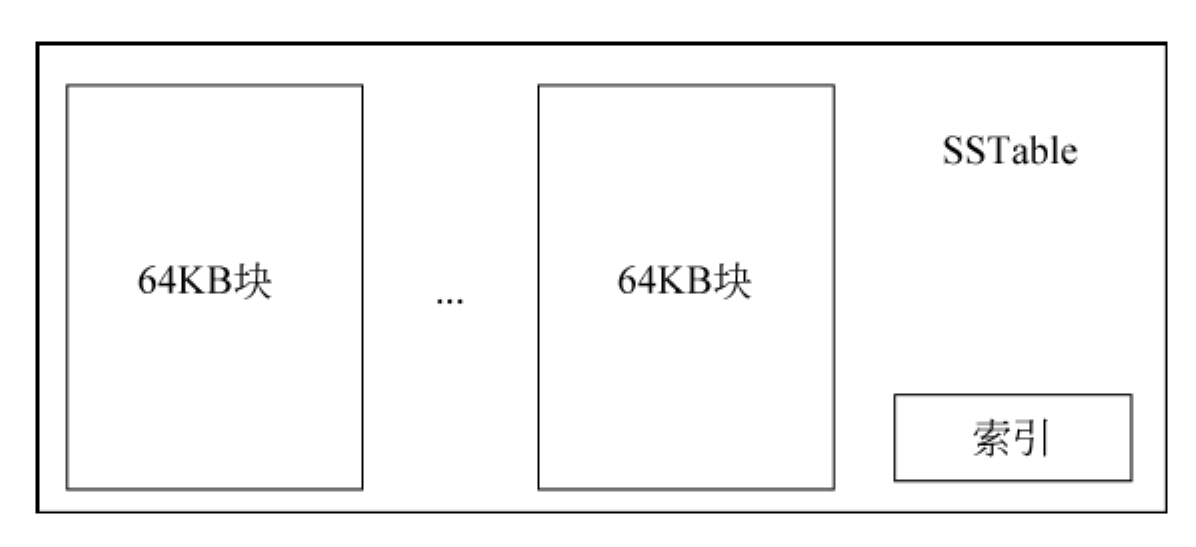


图 10-15　SSTable 结构

SSTable 中的数据被划分成一个个的块(Block)，每个块的大小是可以设置的，一般来说设置为 64KB。在 SSTable 的结尾有一个索引(Index)，这个索引保存了 SSTable 中块的位置信息，在 SSTable 打开时这个索引会被加载进内存，这样用户在查找某个块时首先在内存中查找块的位置信息，然后在硬盘上直接找到这个块，这种查找方法速度非常快。由于每个 SSTable 一般都不是很大，用户还可以选择将其整体加载进内存，这样查找起来会更快。

从概念上来讲子表是表中一系列行的集合，它在系统中的实际组成如图 10-16 所示。

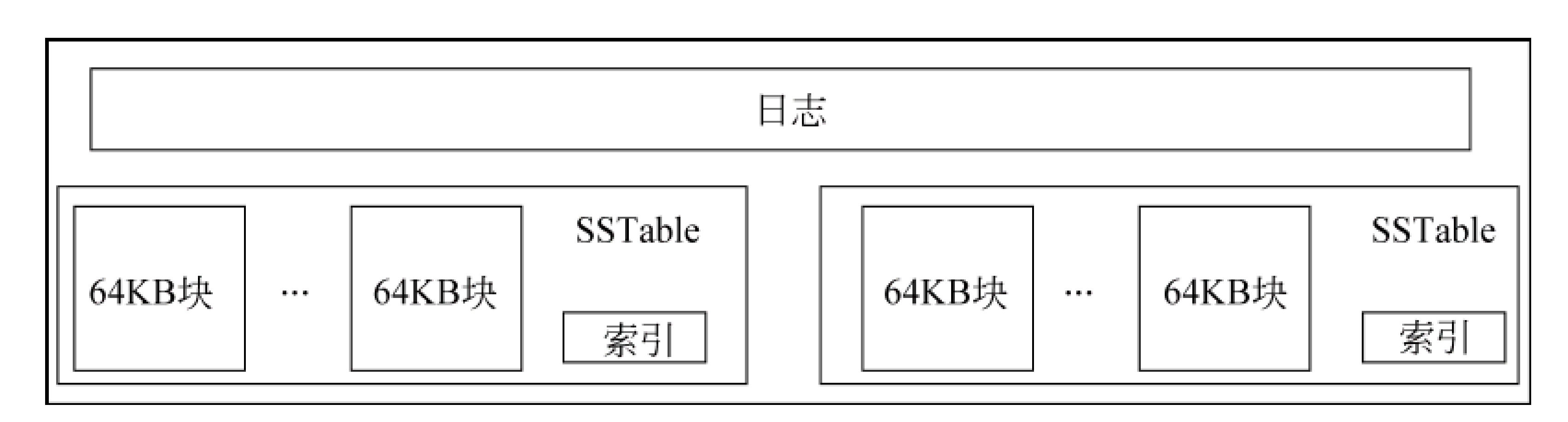


图 10-16　子表实际组成

每个子表都是由多个 SSTable 以及日志(Log)文件构成的。有一点需要注意，那就是不同子表的 SSTable 可以共享，也就是说某些 SSTable 会参与多个子表的构成，而由子表构成的表则不存在子表重叠的现象。Bigtable 中的日志文件是一种共享日志，也就是说系统并不是对子表服务器上每个子表都单独地建立一个日志文件，每个子表服务器上仅保存一个日志文件，某个子表日志只是这个共享日志的一个片段。这样会节省大量的空间，但在恢复时却有一定的难度，因为不同的子表可能会被分配到不同的子表服务器上，一般情况下每个子表服务器都需要读取整个共享日志来获取其对应的子表日志。Google 为了避免这种情况出现，对日志做了一些改进。Bigtable 规定将日志的内容按照键值进行排序，这样不同的子表服务器都可以连续读取日志文件了。一般来说每个子表的大小在 100～200MB 之间。每个子表服务器上保存的子表数量可以从几十到上千不等，通常情况下是 100 个左右。

（2）子表地址

子表地址的查询是经常碰到的操作。在 Bigtable 系统的内部采用的是一种类似 B+树的三层查询体系。子表地址结构如图 10-17 所示。

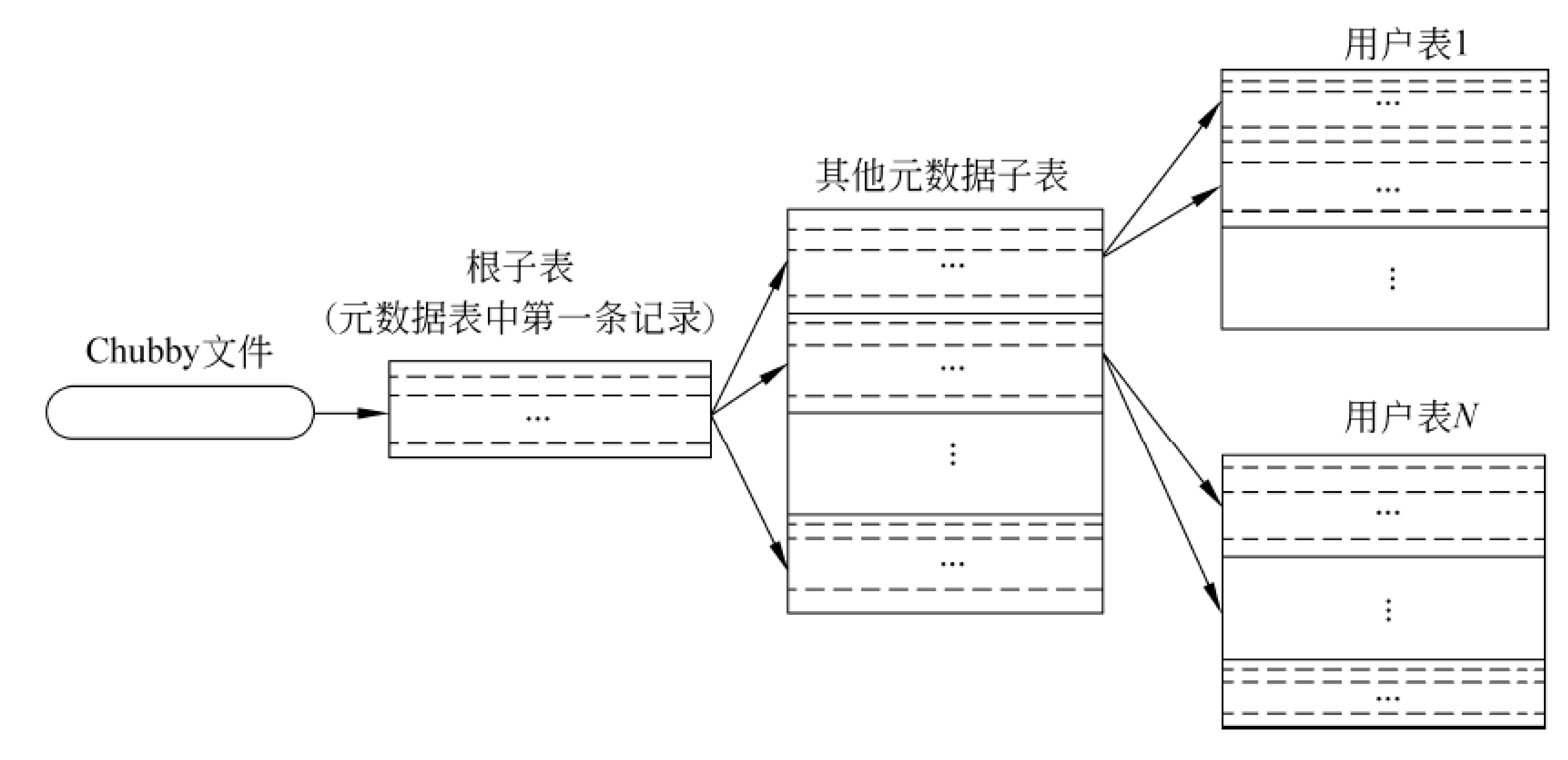


图 10-17　子表地址结构

所有的子表地址都被记录在元数据表中，元数据表也是由一个个的元数据子表（Metadata tablet）组成的。根子表是元数据表中一个比较特殊的子表，它既是元数据表的第一条记录，也包含了其他元数据子表的地址，同时 Chubby 中的一个文件也存储了这个根子表的信息。这样在查询时，首先从 Chubby 中提取这个根子表的地址，进而读取所需的元数据子表的位置，最后就可以从元数据子表中找到待查询的子表。除了这些子表的元数据之外，元数据表中还保存了其他一些有利于调试和分析的信息，比如事件日志等。

为了减少访问开销，提高客户访问效率，Bigtable 使用了缓存（Cache）和预取（Prefetch）技术，这两种技术手段在体系结构设计中是很常用的。子表的地址信息被缓存在客户端，客户在寻址时直接根据缓存信息进行查找。一旦出现缓存为空或缓存信息过时的情况，客户端就需要按照图 10-17 所示方式进行网络的来回通信（Network Round-trips）进行寻址，在缓存为空的情况下需要三个网络来回通信。如果缓存的信息是过时的，则需要六个网络来回通信。其中三个用来确定信息是过时的，另外三个获取新的地址。预取则是在每次访问元数据表时不仅仅读取所需的子表元数据，而是读取多个子表的元数据，这样下次需要时就不用再次访问元数据表。

（3）子表数据存储及读写操作

在数据的存储方面 Bigtable 做出了一个非常重要的选择，那就是将数据存储划分成两块。较新的数据存储在内存中一个称为内存表（Memtable）的有序缓冲里，较早的数据则以 SSTable 格式保存在 GFS 中。这种技术在数据库中不是很常用，但 Google 还是做出了这种选择，实际运行的效果也证明 Google 的选择虽然大胆却是正确的。

从图 10-18 中可以看出读和写操作有很大的差异性。做写操作时，首先查询 Chubby 中保存的访问控制列表确定用户具有相应的写权限，通过认证之后写入的数据首先被保存在提交日志（Commit Log）中。提交日志中以重做记录（Redo Record）的形式保存着最近的一系列数据更改，这些重做记录在子表进行恢复时可以向系统提供已完成的更改信息。数据成功提交之后就被写入内存表中。在做读操作（Read Op）时，首先还是要通过认证，之后

读操作就要结合内存表和 SSTable 文件来进行，因为内存表和 SSTable 中都保存了数据。

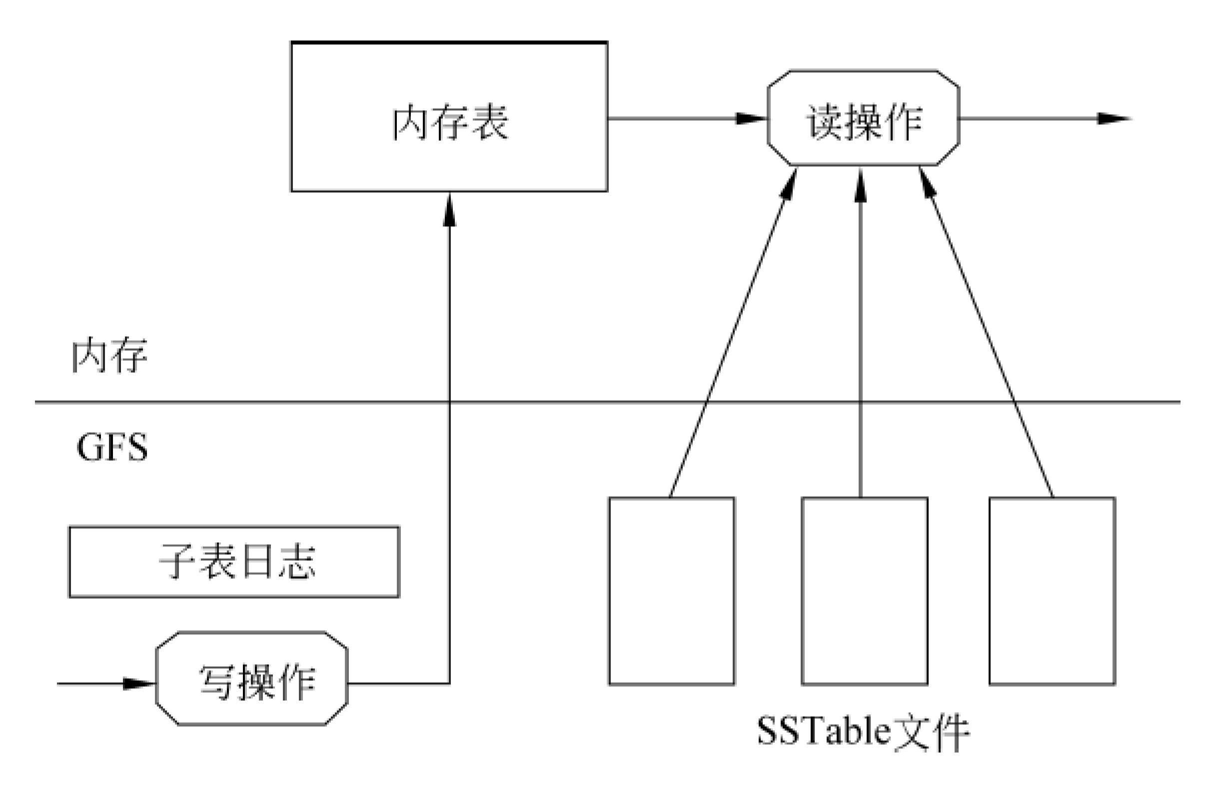


图 10-18　Bigtable 数据存储及读写操作

在数据存储中还有一个重要问题，就是数据压缩的问题。内存表的空间毕竟是很有限的，当其容量达到一个阈值时，旧的内存表就会被停止使用并压缩成 SSTable 格式的文件。在 Bigtable 中有三种形式的数据压缩，分别是次压缩（Minor Compaction）、合并压缩（Merging Compaction）和主压缩（Major Compaction）。三者之间的关系如图 10-19 所示。

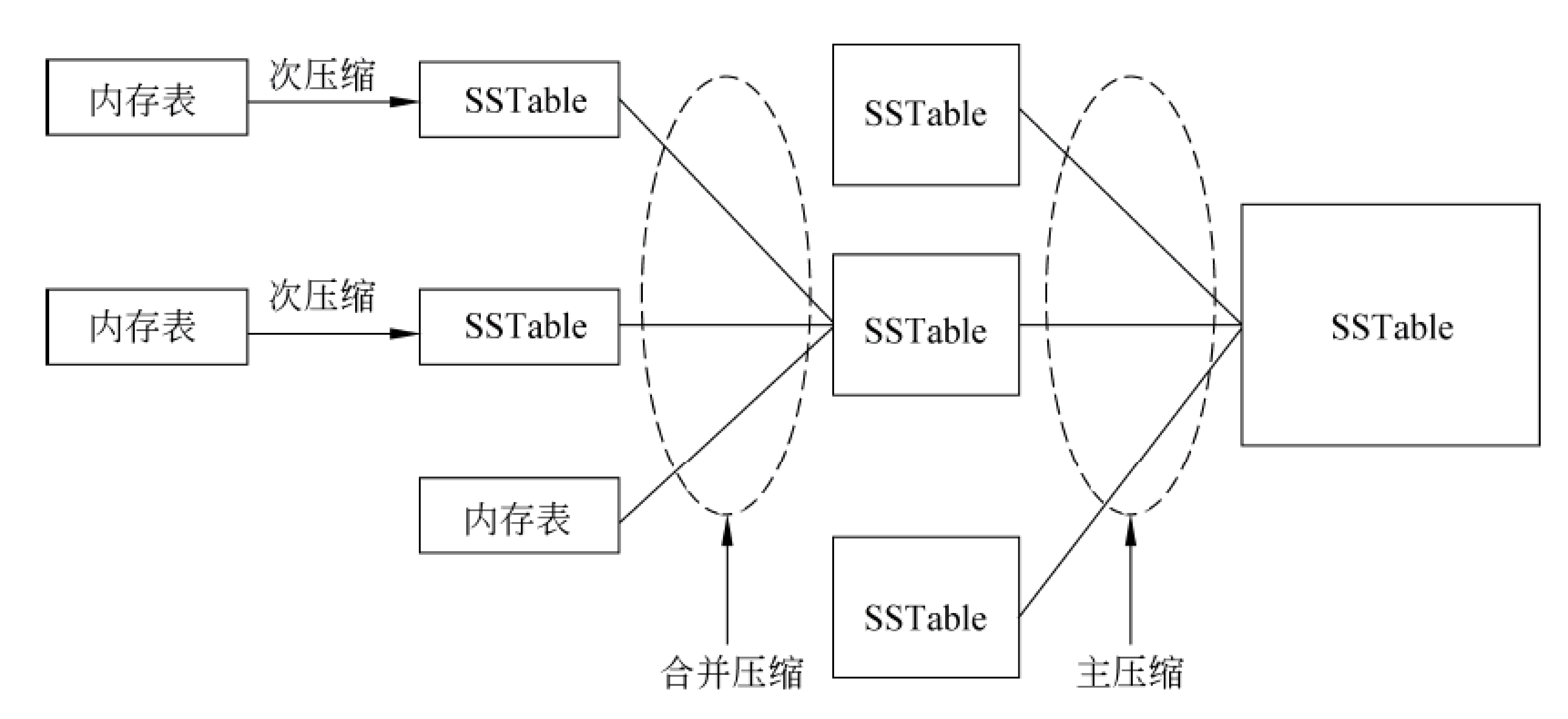


图 10-19　三种形式压缩之间的关系

每一次旧的内存表停止使用时都会进行一个次压缩操作，这会产生一个 SSTable。但如果系统中只有这种压缩，SSTable 的数量就会无限制地增加下去。由于读操作要使用 SSTable，数量过多的 SSTable 显然会影响读的速度。而在 Bigtable 中，读操作实际上比写操作更重要，因此 Bigtable 会定期地执行一次合并压缩的操作，将一些已有的 SSTable 和现有的内存表一并进行一次压缩。主压缩其实是合并压缩的一种，只不过它将所有的 SSTable 一次性压缩成一个大的 SSTable 文件。主压缩也是定期执行的，执行一次主压缩之后可以保证将所有的被压缩数据彻底删除，如此一来，既回收了空间又能保证敏感数据的安全性（因为这些敏感数据被彻底删除了）。

（4）性能优化

上述各种操作已经可以实现 Bigtable 的所有功能了，但是这些基本的功能很多时候并不是很符合用户的使用习惯，或者执行的效率较低。有些功能 Bigtable 自身已经进行了优

化，包括使用缓存、共享式的提交日志以及利用系统的不变性。这些手段在前面已经有了简单的介绍，这里不再讲解。除此之外，Bigtable 还允许用户个人在基本操作基础上对系统进行一些优化。这一部分主要向读者介绍用户可以使用的几个重要优化措施。实际上这些技术手段都是一些已有的数据库方法，只不过 Google 将它具体地应用于 Bigtable 之中罢了。

(5) 局部性群组(Locality groups)

Bigtable 允许用户将原本并不存储在一起的数据以列族为单位根据需要组织在一个单独的 SSTable 中，以构成一个局部性群组。这实际上就是数据库中垂直分区技术的一个应用。在被 Bigtable 保存的网页列关键字中，有的用户可能只对网页内容感兴趣，那么它可以通过设置局部性群组只看内容这一列。有的则会对诸如网页语言、网站排名等可以用于分析的信息比较感兴趣，也可以将这些列设置到一个群组中。局部性群组如图 10-20 所示。

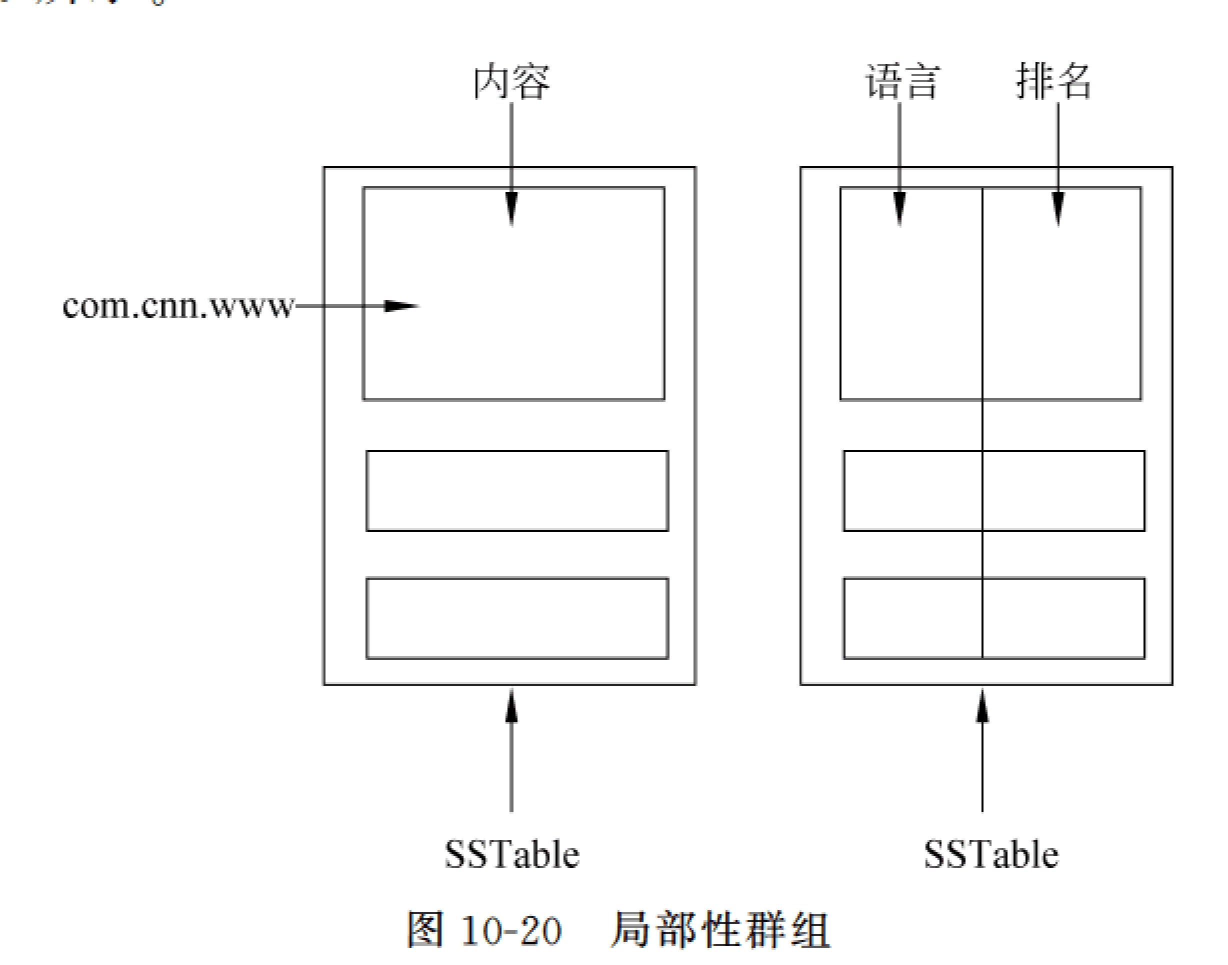


图 10-20　局部性群组

通过设置局部性群组，用户可以只看自己感兴趣的内容，对某个用户来说的大量无用信息无须读取。对于一些较小的且会被经常读取的局部性群组，用户可以将其 SSTable 文件直接加载进内存，这可以明显地改善读取效率。

(6) 压缩

压缩可以有效地节省空间，Bigtable 中的压缩被应用于很多场合。首先压缩可以被用在构成局部性群组的 SSTable 中，可以选择是否对个人的局部性群组的 SSTable 进行压缩。Bigtable 中这种压缩是对每个局部性群组独立进行的，虽然这样会浪费一些空间，但是在需要读时解压速度非常快。通常情况下，用户可以采用两步压缩的方式：第一步利用 Bentley & McIlroy 方式(BMDiff)在大的扫描窗口将常见的长串进行压缩；第二步采取 Zippy 技术进行快速压缩，它在一个 16KB 大小的扫描窗口内寻找重复数据，这个过程非常快。压缩技术还可以提高子表的恢复速度，当某个子表服务器停止使用后，需要将上面所有的子表移至另一个子表服务器来恢复服务。在转移之前要进行两次压缩，第一次压缩减少了提交日志中的未压缩状态，从而减少了恢复时间。在文件正式转移之前还要进行一次压缩，这次压缩主要是将第一次压缩后遗留的未压缩空间进行压缩。完成这两步之后压缩的文件就会被转移至另一个子表服务器。

(7) 布隆过滤器(Bloom Filter)

Bigtable 向用户提供了一种称为布隆过滤器的数学工具。布隆过滤器是巴顿·布隆在1970 年提出的,实际上它是一个很长的二进制向量和一系列随机映射函数,在读操作中确定子表的位置时非常有用。布隆过滤器的速度快,省空间。而且它有一个最大的好处是绝不会将一个存在的子表判定为不存在。不过布隆过滤器也有一个缺点,那就是在某些情况下它会将不存在的子表判断为存在。不过这种情况出现的概率非常小,跟它带来的巨大好处相比,这个缺点是可以忍受的。

包括 Google Analytics、Google Earth、个性化搜索、Orkut 和 RRS 阅读器在内的几十个项目都使用了 Bigtable。这些应用对 Bigtable 的要求以及使用的集群机器数量都是各不相同的,但是从实际运行来看,Bigtable 完全可以满足这些不同需求的应用,而这一切都得益于其优良的构架以及恰当的技术选择。与此同时 Google 还在不断地对 Bigtable 进行一系列的改进,通过技术改良和新特性的加入提高系统的运行效率及稳定性。

任务拓展

1. 什么是云?
2. 为什么要分公有云和私有云?它们有什么特点?
3. 云灾备和普通灾备有何不同?

第 11 章　大数据存储



任务目标

- 了解什么是大数据；
- 了解分布式存储的概念；
- 了解分布式表格系统和分布式数据库；
- 了解大数据存储的关键技术。



项目背景

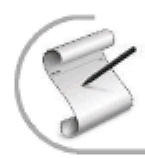
随着云计算、物联网、移动互联网等技术的发展，人类社会各个领域产生的数据量都急剧增长，根据 IDC(International Data Corporation)的统计，2011 年全球被创建和复制的数据总量达 1.8ZB，预计到 2020 年会达到 40ZB，大数据时代已经到来，而且已成为社会各界关注的焦点。大数据的产生源于数据产生方式的改变：随着各种新兴网络服务的出现，服务的内容已经从单一的文本格式转变为各种多媒体形式，如视频、图像、声音等；各种用户产生数据(UGC)，如博客、微博、微信等新型社交媒体中的发展，都增加了用户网络行为数据；物联网传感器网络的广泛使用也自动产生了大量的数据。对于大数据，最具有代表性的是其 4V 的特点如下。

(1) Volume：数据规模大。

(2) Velocity：数据的产生速度很快，处理速度快。

(3) Variety：多样性，包括各种不同的类型和编码格式，数据类型繁多。

(4) Veracity：真实性。



项目描述

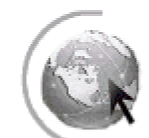
只有真实准确的数据才使得对数据的分析有意义。现如今，大数据已经成为一种重要的基础资源，并形成了以数据为中心、以存储为中心的发展趋势，因此如何存储这些数量骤升的数据已经成为企业和学术界的研究热点。



项目分析

由于数据量的爆炸式增长(特别是非结构化数据每年以 50%的速度增加)、应用规模的扩大以及用户的高并发访问等原因，传统的数据存储系统达到瓶颈，不能胜任大数据环境下的数据的存储、分析和处理工作。首先在数据规模上，传统的数据库已经不能满足高并发、高可靠性的需求；其次大数据环境下要处理的数据种类繁多，包括结构化、非结构化及半结

构化的数据，以处理结构化、同质数据为主的传统数据库系统不能有效处理。作为大数据基础设施的存储系统必须具有高的 IOPS（每秒 I/O 操作次数）和大容量，支持水平扩展（Scale-out）。



项目实现

大数据存储的选择有 NAS（Network Attached Storage）和分布式存储系统。NAS 主要针对文件级别的存储，通过把多个存储节点以网络连接起来来增加存储容量和处理能力。支持横向扩展的 NAS 虽然能够用来处理当今高速增长的数据，但是成本很高，使用特定的文件系统，每当增设磁盘容量的时候就会增加 CPU、内存和网络资源的开销，而且通常 NAS 会使用 RAID-6 来保护数据，更是增加了扩展的成本。所以，分布式存储系统应运而生，其特点是水平高度可扩展、能够容错、高可用、能快速检索出海量数据等。分布式存储按照其存储接口可以分为对象存储、块存储和文件存储。本章展开介绍分布式存储系统和相关的关键技术。

11.1 大数据存储的概念

“大数据”通常指的是那些数量巨大、难于收集、处理、分析的数据集，也指那些在传统基础设施中长期保存的数据。这里的“大”有几层含义，它可以形容组织的大小，而更重要的是，它界定了企业中 IT 基础设施的规模。业内对大数据应用寄予了无限的期望，商业信息积累得越多价值就越大，只不过我们需要一个方法把这些价值挖掘出来。

也许人们对大数据的印象主要从存储容量的廉价性而来，但实际上，企业每天都在创造大量的数据，而且越来越多，而人们正在努力地从浩如烟海的数据中寻觅有价值的商业情报。另一方面，用户还会保存那些已经分析过的数据，因为这些旧数据可以与未来收集的新数据进行对照，依然有潜在的利用可能。

1. 为什么现在需要大数据

与以往相比，我们除了有能力存储更多的数据量之外，还要面对更多的数据类型。这些数据的来源包括网上交易、网络社交活动、自动传感器、移动设备以及科学仪器等。除了那些固定的数据生产源，各种交易行为还可能加快数据的积累速度。比如说，社交类多媒体数据的爆炸性增长就源于新的网上交易和记录行为。数据永远都在增长之中，但是，只有存储海量数据的能力是不够的，因为这并不能保证我们能够成功地从中搜寻出商业价值。

2. 数据是重要的生产要素

信息时代，数据俨然已成为一种重要的生产要素，如同资本、劳动力和原材料等其他要素一样，而且作为一种普遍需求，它也不再局限于某些特殊行业的应用。各行各业的公司都在收集并利用大量的数据分析结果，尽可能地降低成本，提高产品质量，提高生产效率以及创造新的产品。例如，通过分析直接从产品测试现场收集的数据，能够帮助企业改进设计。此外，一家公司还可以通过深入分析客户行为，对比大量的市场数据，从而超越他的竞争对手。

3. 存储技术必须跟上

随着大数据应用的爆发性增长,它已经衍生出了自己独特的架构,而且也直接推动了存储、网络以及计算技术的发展。毕竟处理大数据这种特殊的需求是一个新的挑战。硬件的发展最终还是由软件需求推动的,就这个例子来说,我们很明显地看到大数据分析应用需求正在影响着数据存储基础设施的发展。

从另一方面看,这一变化对存储厂商和其他 IT 基础设施厂商未尝不是一个机会。随着结构化数据和非结构化数据量的持续增长,以及分析数据来源的多样化,此前存储系统的设计已经无法满足大数据应用的需要。存储厂商已经意识到这一点,他们开始修改基于块和文件的存储系统的架构设计以适应这些新的要求。在这里,我们会讨论哪些与大数据存储基础设施相关的属性,看看它们如何迎接大数据的挑战。

4. 容量问题

这里所说的“大容量”通常可达到 PB 级的数据规模,因此,海量数据存储系统也一定要有相应等级的扩展能力。与此同时,存储系统的扩展一定要简便,可以通过增加模块或磁盘柜来增加容量,甚至不需要停机。基于这样的需求,客户现在越来越青睐 Scale-out 架构的存储。Scale-out 集群结构的特点是每个节点除了具有一定的存储容量之外,内部还具备数据处理能力以及互联设备。与传统存储系统的烟囱式架构完全不同,Scale-out 架构可以实现无缝平滑的扩展,避免存储孤岛。

“大数据”应用除了数据规模巨大之外,还意味着拥有庞大的文件数量。因此如何管理文件系统层累积的元数据是一个难题,处理不当会影响到系统的扩展能力和性能,而传统的 NAS 系统就存在这一瓶颈。所幸的是,基于对象的存储架构就不存在这个问题,它可以在一个系统中管理十亿级别的文件数量,而且还不会像传统存储一样遭遇元数据管理的困扰。基于对象的存储系统还具有广域扩展能力,可以在多个不同的地点部署并组成一个跨区域的大型存储基础架构。

5. 延迟问题

“大数据”应用还存在实时性的问题。特别是涉及与网上交易或者金融类相关的应用。举个例子来说,网络成衣销售行业的在线广告推广服务需要实时地对客户的浏览记录进行分析,并准确地进行广告投放。这就要求存储系统必须能够支持上述特性同时保持较高的响应速度,因为响应延迟的结果是系统会推送“过期”的广告内容给客户。这种场景下,Scale-out 架构的存储系统就可以发挥出优势,因为它的每一个节点都具有处理和互联组件,在增加容量的同时处理能力也可以同步增长。而基于对象的存储系统则能够支持并发的数据流,从而进一步提高数据吞吐量。

有很多“大数据”应用环境需要较高的 IOPS 性能,比如 HPC 高性能计算。此外,服务器虚拟化的普及也导致了对高 IOPS 的需求,正如它改变了传统 IT 环境一样。为了迎接这些挑战,各种模式的固态存储设备应运而生,小到简单的在服务器内部做高速缓存,大到全固态介质的可扩展存储系统等都在蓬勃发展。

6. 并发访问

一旦企业认识到大数据分析应用的潜在价值,他们就会将更多的数据集纳入系统进行比较,同时让更多的人分享并使用这些数据。为了创造更多的商业价值,企业往往会综合分析那些来自不同平台下的多种数据对象,包括全局文件系统在内的存储基础设施就能够帮

助用户解决数据访问的问题，全局文件系统允许多个主机上的多个用户并发访问文件数据，而这些数据则可能存储在多个地点的多种不同类型的存储设备上。

7. 安全问题

某些特殊行业的应用，比如金融数据、医疗信息以及政府情报等都有自己的安全标准和保密性需求。虽然对于 IT 管理者来说这些并没有什么不同，而且都是必须遵从的，但是，大数据分析往往需要多类数据相互参考，而在过去并不会有这种数据混合访问的情况，因此大数据应用也催生出一些新的、需要考虑的安全性问题。

成本问题“大”，也可能意味着代价不菲。而对于那些正在使用大数据环境的企业来说，成本控制是关键的问题。想控制成本，就意味着我们要让每一台设备都实现更高的“效率”，同时还要减少那些昂贵的部件。目前，像重复数据删除等技术已经进入到主存储市场，而且现在还可以处理更多的数据类型，这都可以为大数据存储应用带来更多的价值，提升存储效率。在数据量不断增长的环境中，通过减少后端存储的消耗，哪怕只是降低几个百分点，都能够获得明显的投资回报。此外，自动精简配置、快照和克隆技术的使用也可以提升存储的效率。

很多大数据存储系统都包括归档组件，尤其对那些需要分析历史数据或需要长期保存数据的机构来说，归档设备必不可少。从单位容量存储成本的角度看，磁带仍然是最经济的存储介质，事实上，在许多企业中，使用支持 TB 级大容量磁带的归档系统仍然是事实上的标准和惯例。

对成本控制影响最大的因素是那些商业化的硬件设备。因此，很多初次进入这一领域的用户以及那些应用规模最大的用户都会定制他们自己的“硬件平台”而不是用现成的商业产品，这一举措可以用来平衡他们在业务扩展过程中的成本控制战略。为了适应这一需求，现在越来越多的存储产品都提供纯软件的形式，可以直接安装在用户已有的、通用的或者现成的硬件设备上。此外，很多存储软件公司还在销售以软件产品为核心的软硬一体化装置，或者与硬件厂商结盟，推出合作型产品。

8. 数据的积累

许多大数据应用都会涉及法规遵从问题，这些法规通常要求数据要保存几年或者几十年。比如医疗信息通常是为了保证患者的生命安全，而财务信息通常要保存 7 年。而有些使用大数据存储的用户却希望数据能够保存更长的时间，因为任何数据都是历史记录的一部分，而且数据的分析大都是基于时间段进行的。要实现长期的数据保存，就要求存储厂商开发出能够持续进行数据一致性检测的功能以及其他保证长期高可用的特性。同时还要实现数据直接在原位更新的功能需求。

9. 灵活性

大数据存储系统的基础设施规模通常都很大，因此必须经过仔细设计，才能保证存储系统的灵活性，使其能够随着应用分析软件一起扩容及扩展。在大数据存储环境中，已经没有必要再做数据迁移了，因为数据会同时保存在多个部署站点。一个大型的数据存储基础设施一旦开始投入使用，就很难再调整了，因此它必须能够适应各种不同的应用类型和数据场景。

10. 应用感知

最早一批使用大数据的用户已经开发出了一些针对应用的定制的基础设施，比如针对

政府项目开发的系统，还有大型互联网服务商创造的专用服务器等。在主流存储系统领域，应用感知技术的使用越来越普遍，它也是改善系统效率和性能的重要手段，所以，应用感知技术也应该用在大数据存储环境里。

小用户怎么办？依赖大数据的不仅仅是那些特殊的大型用户群体，作为一种商业需求，小型企业未来也一定会应用到大数据。我们看到，有些存储厂商已经在开发一些小型的“大数据”存储系统，主要吸引那些对成本比较敏感的用户。

11.2 分布式存储系统介绍

分布式存储系统包含多个自主的处理单元，通过计算机网络互联来协作完成分配的任务。分布式存储系统更能适应现在分布广泛的企业的组织结构，更加可靠，响应速度更快；当今很多的互联网应用在本质上就是分布式的，例如基于 Web 的应用、电子商务、广告推送、在线游戏、生产控制系统等；分布式架构通过分而治之的策略能够更好地处理当今我们面临的大规模数据处理问题，这也是其能够得到广泛部署的根本原因。分布式存储系统的目的在于在多个节点上进行数据存储和管理，对外作为一个整体提供服务。分布式存储系统作为底层管理数据的基础设施，让分布式处理更加简单和高效。分布式存储的研究有三十多年，出现了很多种不同的系统，根据系统中数据存储格式和存取接口可以把分布式存储系统分为分布式文件系统、分布式键值系统、分布式表格系统以及分布式数据库，本文将分别介绍几种典型的分布式存储系统。

11.2.1 分布式文件系统

大数据存储管理需要多种技术的协同工作，其中文件系统为其提供最底层存储能力的支持。分布式文件系统(Distributed File System，DFS)是一个基于 C/S 的应用程序，允许来自不同终端的用户访问和处理服务器上的文件。DFS 的实现有很多，如 NFS、Andrew File System、Coda 等，其中最著名的是 Google 文件系统(Google File System，GFS)。它构建在大量普通的廉价设备之上，支持自动容错；主要针对文件较大、读操作远大于写操作的应用场景，GFS 把大文件划分为 64MB 的数据块(Chunk)；采用主从(Master-Slave)结构，主控服务器用来实现元数据管理、副本管理、自动负载均衡、记录操作日志等操作。很多其他分布式文件系统都借鉴了 GFS 的思想，如淘宝文件系统、Facebook Haystack 等。

本地文件系统如 ext3、reiserfs 等(这里不讨论基于内存的文件系统)，它们管理本地的磁盘存储资源、提供文件到存储位置的映射，并抽象出一套文件访问接口供用户使用。但随着互联网企业的高速发展，这些企业对数据存储的要求越来越高，而且模式各异，如淘宝主站的大量商品图片，其特点是文件较小，但数量巨大；而类似于 youtube、优酷这样的视频服务网站，其后台存储着大量的视频文件，容量大多在数十兆到数吉字节不等。这些应用场景都是传统文件系统不能解决的。分布式文件系统将数据存储在物理上分散的多个存储节点上，对这些节点的资源进行统一的管理与分配，并向用户提供文件系统访问接口，其主要解决了本地文件系统在文件大小、文件数量、打开文件数等方面的限制问题。

11.2.2　典型架构

目前比较主流的一种分布式文件系统架构如图 11-1 所示，通常包括主控服务器（或称元数据服务器、名字服务器等，一般会配置备用主控服务器以便在故障时接管服务，也可以两个都为主的模式），多个数据服务器（或称存储服务器、存储节点等），以及多个客户端，客户端可以是各种应用服务器，也可以是终端用户。

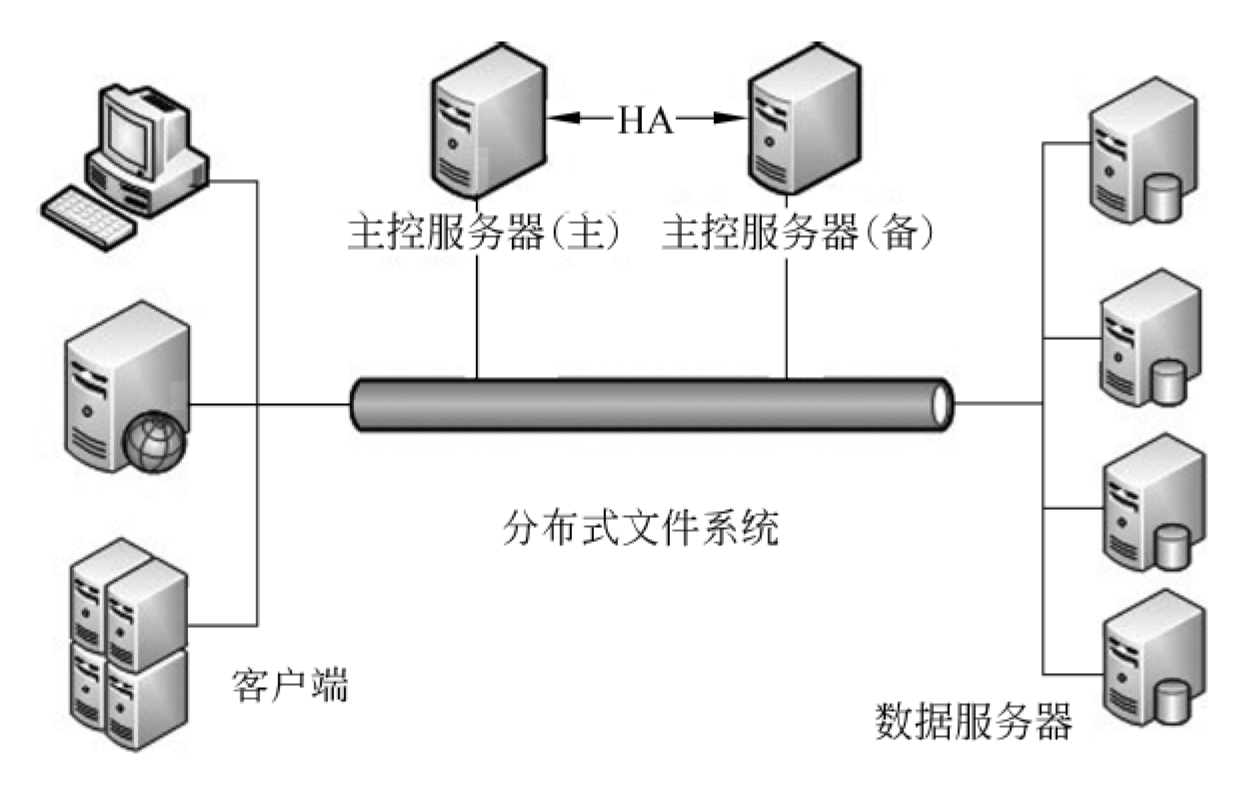


图 11-1　分布式文件系统架构

分布式文件系统的数据存储解决方案，归根结底是将大问题划分为小问题。大量的文件均匀分布到多个数据服务器上后，每个数据服务器存储的文件数量就少了，另外通过使用大文件存储多个小文件的方式，总能把单个数据服务器上存储的文件数降到单机能解决的规模。对于很大的文件，将大文件划分成多个相对较小的片段，存储在多个数据服务器上（目前，很多本地文件系统对超大文件的支持已经不存在问题了，如 ext3 文件系统使用 4KB 块时，文件最大能到 4TB，ext4 则能支持更大的文件，只是受限于磁盘的存储空间）。

理论上，分布式文件系统可以只由客户端和多个数据服务器组成，客户端根据文件名决定将文件存储到哪个数据服务器，但一旦有数据服务器失效时，问题就变得复杂，客户端并不知道数据服务器宕机的消息，仍然连接它进行数据存取，导致整个系统的可靠性极大地降低，而且完全有客户端决定数据分配是非常不灵活的，其不能根据文件特性制定不同的分布策略。

数据服务器的状态管理可分为分散式和集中式两种方式，前者是让多个数据服务器相互管理，如每个服务器向其他所有的服务器发送心跳信息，但这种方式开销较大，控制不好容易影响到正常的数据服务，而且工程实现较为复杂；后者是指通过一个独立的服务器（如图 11-1 中的主控服务器）来管理数据服务器，每个服务器向其汇报服务状态来达到集中管理的目的，这种方式简单易实现，目前很多分布式文件系统都采用这种方式，如 GFS、TFS、MooseFS 等。主控服务器在负载较大时会出现单点，较多的解决方案是配置备用服务器，以便在故障时接管服务，如果需要，主备之间需要进行数据的同步。

11.2.3　问题及解决方法

本小节主要讨论基于图 11-1 架构的分布式文件系统的相关原理，工程实现时需要解决的问题和解决问题的基本方法。分布式文件系统涉及的主要问题及解决方法如图 11-2 所示。为方便描述，以下将主控服务器简称 Master，数据服务器简称 DS(Data Server)。

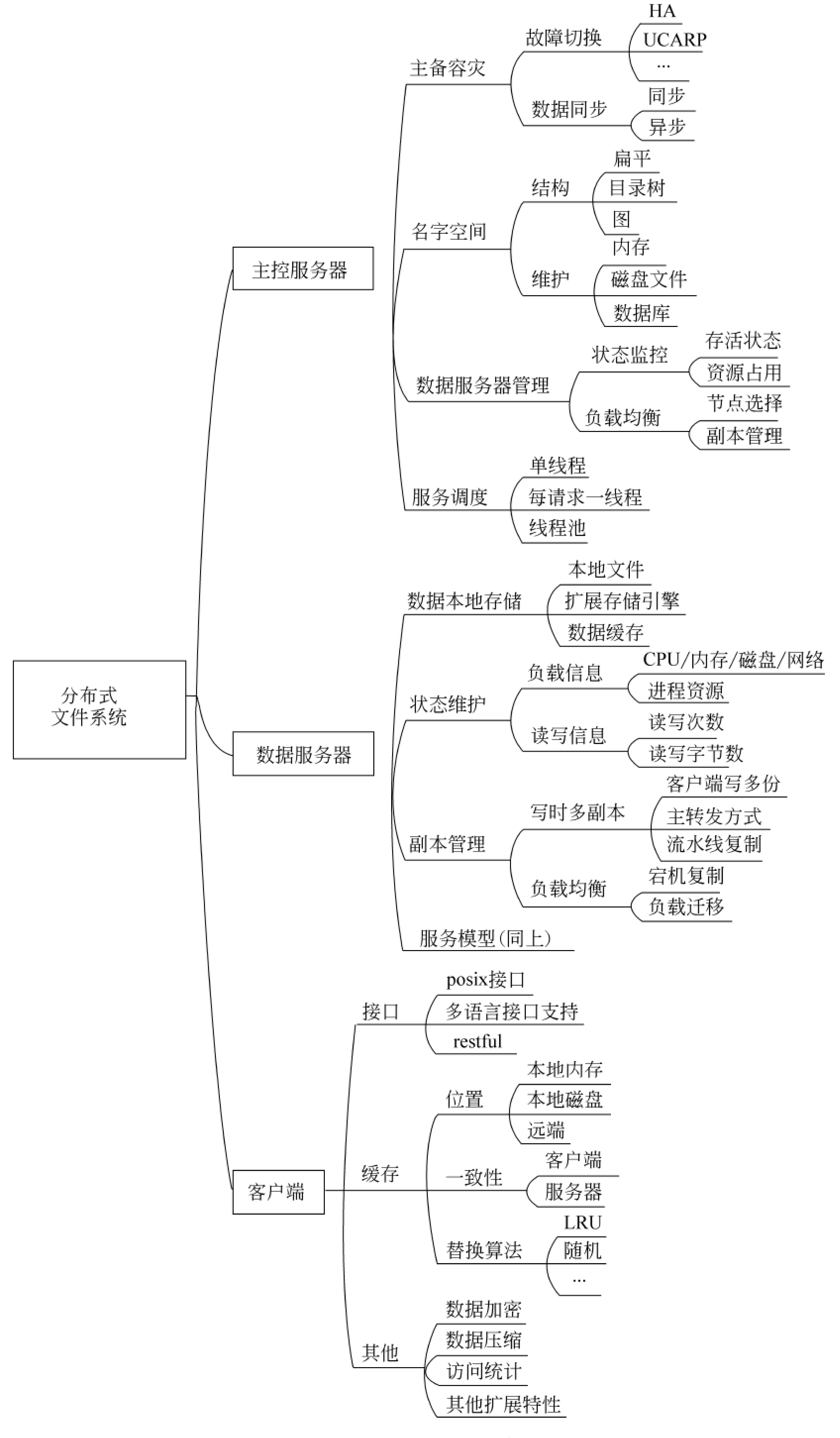


图 11-2　主要问题及解决方法

11.2.4　主控服务器

1. 命名空间的维护

Master 负责维护整个文件系统的命名空间，并暴露给用户使用，命名空间的结构主要有典型目录树结构如 MooseFS 等，扁平化结构如淘宝 TFS（目前已提供目录树结构支持），图结构（主要面向终端用户，方便用户根据文件关联性组织文件，只在论文中看到过）。

为了维护名字空间，需要存储一些辅助的元数据如文件（块）到数据服务器的映射关系，文件之间的关系等，为了提升效率，很多文件系统采取将元数据全部内存化（元数据通常较小）的方式如 GFS、TFS。有些系统则借助数据库来存储元数据如 DBFS，还有些系统则采用本地文件来存储元数据如 MooseFS。

一种简单的实现目录树结构的方式是，在 Master 上存储与客户端完全一样的命名空间，对应的文件内容为该文件的元数据，并通过在 Master 上采用 ReiserFS 来进行小文件存储优化，对于大文件的存储（文件数量不会成为 Master 的瓶颈），这种方式简单易实现。曾经参与的 DNFS 系统的开发就是使用这种方式，DNFS 主要用于存储视频文件，视频数量在百万级别，Master 采用这种方式时，文件数量上不会成为瓶颈。

2. 数据服务器管理

除了维护文件系统的命名空间，Master 还需要集中管理数据 DS，可通过轮询 DS 或由 DS 报告的方式实现。在接收到客户端的请求时，Master 需要根据各个 DS 的负载等信息选择一组（根据系统配置的副本数）DS 为其服务；当 Master 发现有 DS 宕机时，需要对一些副本数不足的文件（块）执行复制计划；当有新的 DS 加入集群或是某个 DS 上负载过高，Master 也可根据需要执行一些副本迁移计划。

如果 Master 的元数据存储是非持久化的，则在 DS 启动时还需要把自己的文件（块）信息汇报给 Master。在分配 DS 时，基本的分配方法有随机选取、RR 轮转、低负载优先等，还可以将服务器的部署作为参考（如 HDFS 分配的策略），也可以根据客户端的信息，将分配的 DS 按照与客户端的远近排序，使得客户端优先选取离自己近的 DS 进行数据存取。

3. 服务调度

Master 最终的目的还是要服务好客户端的请求，除了一些周期性线程任务外，Master 需要服务来自客户端和 DS 的请求，通常的服务模型包括单线程、每请求一线程、线程池（通常配合任务队列）。单线程模型下，Master 只能顺序地服务请求，该方式效率低，不能充分利用好系统资源；每请求一线程的方式虽能并发地处理请求，但由于系统资源的限制，导致创建线程数存在限制，从而限制同时服务的请求数量，另外，线程太多，线程间的调度效率也是个大问题；线程池的方式目前使用较多，通常由单独的线程接受请求，并将其加入到任务队列中，而线程池中的线程则从任务队列中不断地取出任务进行处理。

4. 主备（主）容灾

Master 在整个分布式文件系统中的作用非常重要，其维护文件（块）到 DS 的映射、管理所有的 DS 状态并在某些条件触发时执行负载均衡计划等。为了避免 Master 的单点问题，通常会为其配置备用服务器，以保证在主控服务器节点失效时接管其工作。通常的实现方式是通过 HA、UCARP 等软件为主备服务器提供一个虚拟 IP 提供服务，当备用服务器检测到主宕机时，会接管主服务器的资源及服务。

如果 Master 需要持久化保存一些数据，则需要将数据同步到备用 Maste 上。对于元数据内存化的情况，为了加速元数据的构建，有时也需将主服务器上的操作同步到备用 Master 上。处理方式可分为同步和异步两种。同步方式将每次请求同步转发至备用 Master 上，这样理论上主备服务器时刻保持一致的状态，但这种方式会增加客户端的响应延迟(在客户端对响应延迟要求不高时可使用这种方式)。当备用 Master 宕机时，可不做任何处理，等备用 Master 起来后再同步数据，或是暂时停止写服务，管理员介入启动备用 Master 再正常服务(需业务能容忍)。异步方式则是先暂存客户端的请求信息(如追加至操作日志)，后台线程重放日志到备用 Master 上，这种方式会使得主备服务器的数据存在不一致的情况，具体策略需针对需求制定。

11.2.5 数据服务器

1. 数据本地存储

数据服务器负责文件数据在本地的持久化存储，最简单的方式是将客户每个文件数据分配到一个单独的 DS 上作为一个本地文件存储，但这种方式并不能很好地利用分布式文件系统的特性，很多文件系统使用固定大小的块来存储数据，如 GFS、TFS、HDFS，典型的块大小为 64MB。

对于小文件的存储，可以将多个文件的数据存储在一个块中，并为块内的文件建立索引，这样可以极大地提高存储空间利用率。Facebook 用于存储照片的 HayStack 系统的本地存储方式为，将多个图片对象存储在一个大文件中，并为每个文件的存储位置建立索引，其支持文件的创建和删除，不支持更新(通过删除和创建完成)，新创建的图片追加到大文件的末尾并更新索引，文件删除时，简单的设置文件头的删除标记，系统在空闲时会对大文件进行压缩，把设置删除标记且超过一定时限的文件存储空间回收(延迟删除策略)。淘宝的 TFS 系统采用了类似的方式，对小文件的存储进行了优化，TFS 使用扩展块的方式支持文件的更新。对小文件的存储也可直接借助一些开源的 KV 存储解决方案，如 Tokyo Cabinet (HDB、FDB、BDB、TDB)、Redis 等。

对于大文件的存储，则可将文件存储到多个块上，多个块所在的 DS 可以并行服务，这种需求通常不需要对本地存储做太多优化。

2. 状态维护

DS 除了简单的存储数据外，还需要维护一些状态，首先它需要将自己的状态以心跳包的方式周期性地报告给 Master，使得 Master 知道自己是否正常工作，通常心跳包中还会包含 DS 当前的负载状况(CPU、内存、磁盘 I/O、磁盘存储空间、网络 I/O 等、进程资源，视具体需求而定)，这些信息可以帮助 Master 更好地制定负载均衡策略。

很多分布式文件系统如 HDFS 在外围提供一套监控系统，可以实时地获取 DS 或 Master 的负载状况，管理员可根据监控信息进行故障预防。

3. 副本管理

为了保证数据的安全性，分布式文件系统中的文件会存储多个副本到 DS 上，写多个副本的方式主要分为三种。最简单的第一种方式是客户端分别向多个 DS 写同一份数据，如 DNFS 采用这种方式；第二种方式是客户端向主 DS 写数据，主 DS 向其他 DS 转发数据，如 TFS 采用这种方式；第三种方式采用流水复制的方式，客户向某个 DS 写数据，该 DS 向副

本链中下一个 DS 转发数据，依次类推，如 HDFS、GFS 采取这种方式。

当有节点宕机或节点间负载极不均匀的情况下，Master 会制定一些副本复制或迁移计划，而 DS 实际执行这些计划，将副本转发或迁移至其他的 DS。DS 也可提供管理工具，在需要的情况下由管理员手动执行一些复制或迁移计划。

4. 客户端

(1) 接口

用户最终通过文件系统提供的接口来存取数据，Linux 环境下，最好莫过于能提供 POSIX 接口的支持，这样很多应用(各种语言皆可，最终都是系统调用)能不加修改的将本地文件存储替换为分布式文件存储。

要想文件系统支持 POSIX 接口，一种方式时按照 VFS 接口规范实现文件系统，这种方式需要文件系统开发者对内核有一定的了解；另一种方式是借助 FUSE 软件，在用户态实现文件系统并能支持 POSIX 接口，但是用该软件包开发的文件系统会有额外的用户态、内核态的切换及数据拷贝过程，从而导致其效率不高。

如果不能支持 POSIX 接口，则为了支持不同语言的开发者，需要提供多种语言的客户端支持，如常用的 C/C++ 、Java、PHP、Python 客户端。使用客户端的方式较难处理的一种情况时，当客户端升级时，使用客户端接口的应用要使用新的功能，也需要进行升级，当应用较多时，升级过程非常麻烦。目前一种趋势是提供 Restful 接口的支持，使用 HTTP 协议的方式给应用(用户)访问文件资源，这样就避免功能升级带来的问题。

另外，在客户端接口的支持上，也需根据系统需求权衡，比如 write 接口，在分布式实现上较麻烦，很难解决数据一致性的问题，应该考虑能否只支持 create(update 通过 delete 和 create 组合实现)，或折中支持 append，以降低系统的复杂性。

(2) 缓存

分布式文件系统的文件存取，要求客户端先连接 Master 获取一些用于文件访问的元信息，这一过程一方面加重了 Master 的负担，另一方面增加了客户端请求的响应延迟。为了加速该过程，同时减小 Master 的负担，可将元信息进行缓存，数据可根据业务特性缓存在本地内存或磁盘，也可缓存在远端的 cache 系统上，如淘宝的 TFS 可利用 tair 作为缓存(减小 Master 负担、降低客户端资源占用)。

维护缓存需考虑如何解决一致性问题及缓存替换算法，一致性的维护既可由客户端也可由服务器完成，一种方式是客户端周期性地使 cache 失效或检查 cache 的有效性(需业务上能容忍)，或由服务器在元数据更新后通知客户端使 cache 失效(需维护客户端状态)。使用得较多的替换算法有 LRU、随机替换等。

(3) 其他

客户端还可以根据需要支持一些扩展特性，如将数据进行加密来保证数据的安全性、将数据进行压缩后降低存储空间的使用，或是在接口中封装一些访问统计行为，以支持系统对应用的行为进行监控和统计。

11.2.6　HDFS 介绍

Hadoop 分布式文件系统(HDFS)是运行在通用硬件上的分布式文件系统。HDFS 提供了一个高度容错性和高吞吐量的海量数据存储解决方案。HDFS 已经在各种大型在线服

务和大型存储系统中得到广泛应用,已经成为各大网站等在线服务公司的海量存储事实标准,多年来为网站客户提供了可靠高效的服务。

随着信息系统的快速发展,海量的信息需要可靠存储的同时,还能被大量的使用者快速地访问。传统的存储方案已经从构架上越来越难以适应近几年来的信息系统业务的飞速发展,成为业务发展的瓶颈和障碍。

HDFS 通过一个高效的分布式算法,将数据的访问和存储分布在大量服务器之中,在可靠地多备份存储的同时,还能将访问分布在集群中的各个服务器之上,是传统存储构架的一个颠覆性的发展。HDFS 可以提供以下特性:

① 可自我修复的分布式文件存储系统;

② 高可扩展性,无须停机动态扩容;

③ 高可靠性,数据自动检测和复制;

④ 高吞吐量访问,消除访问瓶颈;

⑤ 使用低成本存储和服务器构建。

11.2.7 分布式文件系统 HDFS 的特性

1. 高吞吐量访问

HDFS 的每个数据块分布在不同机架的一组服务器之上,在用户访问时,HDFS 将会计算使用网络最近的和访问量最小的服务器给用户提供访问。由于数据块的每个拷贝都能提供给用户访问,而不是从单数据源读取,HDFS 对于单数据块的访问将是传统存储方案的数倍。

对于一个较大的文件,HDFS 将文件的不同部分存放于不同服务器之上。在访问大型文件时,系统可以并行从服务器阵列中的多个服务器并行读入,增加了大文件读入的访问带宽。

通过以上实现,HDFS 通过分布式计算的算法,将数据访问均匀分配到服务器阵列中的每个服务器的多个数据拷贝之上,单个硬盘或服务器的吞吐量限制都可以有数倍甚至数百倍的突破,提供了极高的数据吞吐量。

2. 无缝容量扩充

HDFS 将文件的数据块分配信息存放在 NameNode 服务器之上,文件数据块的信息分布存放在 DataNode 服务器上。当整个系统容量需要扩充时,只需要增加 DataNode 的数量,系统会自动地实时将新的服务器匹配进整体阵列之中。之后,文件的分布算法会将数据块搬迁到新的 DataNode 之中,不需任何系统宕机维护或人工干预。通过以上实现,HDFS 可以做到在不停止服务的情况下实时地加入新的服务器作为分布式文件系统的容量升级,不需要人工干预文件的重新分布。

3. 高度容错

HDFS 文件系统假设系统故障(服务器、网络、存储故障等)是常态,而不是异常。因此通过多方面保证数据的可靠性。数据在写入时被复制多份,并且可以通过用户自定义的复制策略分布到物理位置不同的服务器上;数据在读写时将自动进行数据的校验,一旦发现数据校验错误将重新进行复制;HDFS 系统在后台自动连续地检测数据的一致性,并维持数据的副本数量在指定的复制水平上。

11.3　分布式数据库

11.3.1　分布式数据库结构

分布式数据库的典型定义是：分布式数据库是一个数据集合，这些数据在逻辑上属于同一个系统，但物理上却分散在计算机网络的若干站点上，并且要求网络的每个站点具有自治的处理能力，能执行本地的应用。每个站点的计算机还至少参与一个全局应用的执行。

所谓全局应用，要求使用通信子系统在几个站点存取数据。这个定义强调了分布式数据库的两个重要特点：分布性和逻辑相关性。

1. DDBMS 的工作原理

DDBMS(Distributed Database Management System，分布式数据库管理系统)是分布式数据库的核心部分，就其性质可分为匀质和异质两种。若每个站点的 DDBMS 相同，则是匀质的；若至少有两个 DDBMS 不同，则是异质的。

异质 DDBMS 要在不同 DDBMS 的不同数据模型间进行转换，因而比匀质 DDBMS 更复杂。一般来说，若从头开始研制一个 DDBMS，则选择匀质较方便，且通常都选用关系模型。这是由于关系模型易于分布管理。但若 DDBMS 是建立在已有的若干数据库之上，则这些数据库很可能有的是基于关系模型的，有的是基于层次或网络模型的，即它们是不同质的，因此要建立异质的 DDBMS。

用户处理器根据外模式和概念模式把用户命令翻译成格式更适合于机器的规范化命令，并实施完整性约束，同时它负责将规范化格式的数据转换成用户结果格式。

数据处理器根据概念模式和内模式选择通向物理数据的最优或近乎最优的访问路径，用规范化命令翻译成物理命令，并行执行物理命令，再送回结果。它还负责将物理数据转换成规范化数据。这一部分功能通常由 DDBMS 完成。

DDBMS 主要由全局数据库控制和通信系统的组成。分解器根据分布模式负责将来自用户处理器的规范化命令翻译成一个分布执行策略，指明各数据处理器应完成的命令。分布式执行监测器负责分布策略的正确执行以及保证分布环境中事务的原子性。通信子系统负责在网络的各计算面站点间传输命令和数据。局部执行监视器负责监视执行分布策略中的本地部分。合并器把来自几个数据处理器的结果组合起来，形成最终结果。

各站点计算机的自治程度也是分布式数据库系统的一个重要特性。它与分解器和分布执行监视器所处的位置密切相关。若分解器和分布式执行监视器处在外部模式以上，这时分布模式实际只存在于程序员的脑海之中，是一种由程序员来联系的系统。若它们处在外部模式以下概念模式以上，则是联邦式 DDBMS。若它们处在概念模式以下内部模式以上，则称为紧耦合的分布式 DDBMS。若它们在内模式以下，则是集中式 DDBMS 的体系结构。分解器和分布式执行监视器所处的位置越往下，站点的自治能力越弱。

2. DDBMS 技术

分布式数据库设计包括外模式、全局模式、分段模式、分配模式和各站点内模式的定义。

设计时应考虑到下述目标：提高处理的本地性以提高响应速度和降低处理成本；提高数据的可用性和可靠性；合理地分布工作负荷以提高并行处理的能力等。

关系分段的方法有水平分段(又分为简单水平分段和诱导水平分段)、垂直分段和混合分段。设计分段时应该遵守完整性约束规则、重构性条件规则和不相交性条件规则。

复制副本的方法可以提高数据的可用性和可靠性以及处理的局部性。但对副本要加以控制,使得对用户是透明的,即对用户来说好像只存在一个副本一样。一般来说,查询只需对一个副本进行;而更新则要对所有副本进行,这使得更新的算法变得复杂。

3. 目录管理

分布式数据库的目录中存放了系统为保证正确、有效地访问数据而要使用的全部信息。一般应包括各级模式的描述、访问方法的描述、关于数据库的统计数据和一致性信息等。系统根据这些信息将用户查询转换为物理数据库上的查询,并进行查询优化,选择一条最佳的存取路径进行事务管理,以及进行安全性和完整性检查等。

目录的管理有多种方式。最基本的三种方式是:集中式目录,只有一个站点存放目录;全复制目录,每个站点都存放目录;局部目录,每个站点只含有本地数据库的目录。可以把目录本身作为一个特殊的系统库来对待,则对它也有检索、更新、并发控制等访问和维护的问题。

4. 查询处理及其优化

这是分布式数据库的主要技术问题之一。分布式查询处理负责将用户查询转换为各站点上的物理查询,并负责合并各子查询结果以得到最后结果。

查询的效率问题是个关键。因此对用户查询程序应加以优化,寻找一种使查询费用最少的执行策略,这个费用等于传输费用和局部费用之和。有时局部费用相对于传输费用可以忽略不计,如大型广域网联接的系统。但在高速的局域网中,局部费用也必须考虑。

11.3.2 分布式并发控制技术

事务是一个应用或一个应用的一部分,它是数据库的一致性单位,即在事务的执行前后数据库都必须是一致的。分布事务管理必须保证事务具有原子性、持久性、可串行性和隔离性。在大多数系统中是这样得到的,即在现有的本地事务管理程序上面用两阶段托付协议来获得可靠性,用两阶段锁定来进行并发控制,以及用超时来解决死锁检测。

两阶段托付协议确保同一事务的子事务全部托付或全部终止,不管有无发生故障;两阶段托付在不丢失运行记录信息的情况下可从任何故障恢复。两阶段锁定机构要求全部子事务在生长阶段获得锁定而在收缩阶段释放锁定。用于死锁检测的超时机构在那些事务可能因死锁而等待时简单地使它们终止。

并发控制是分布式事务管理的基本任务之一,其目的是保证分布式数据库系统中多个事务的并发、高效及正确地执行。并发控制用来保证事务的可串行性,即事务的并发执行等价于它们按某种次序的串行执行,从而为用户提供了并发透明性。

目前已提出了大量的并发控制方法,最基本的有封锁方法、时间印方法和乐观方法等。时间印方法和乐观方法理论上研究得较多,大多数原型 DDBMS 和商品化 DDBMS 采用二段锁方法。

1. 封锁法

对于由于共享而产生的冲突,一般可采用封锁技术。即在对数据操作之前,先向并发控制机构提出封锁申请,如果不与其他事务发生冲突,申请被批准,系统对数据加上锁标志;否

则令其等待，直至其他事务释放对此数据的封锁。

封锁方法中的一个重要问题是封锁粒度的选择。理想的情况是只封锁与操作有关的数据集，通常称为完整性相关域。为了保持数据完整性，至少应封锁完整性相关域中的数据，从而使其中任何施加于现有数据集合的操作所影响的数据都置于封锁之中。封锁分为读锁和写锁，也称共享锁和排他锁。当事务出现冲突操作时，通过加锁原则及锁的相容性机制实现。

冲突操作的可串行化调度：当数据共享时，事务并行执行；当数据排它时事务串行执行。使用锁模型实现并发的最著名算法是两段锁协议。其基本方法是任何事务对数据项操作之前先加锁。加锁的方法是，在事务中的全部加锁操作在第一次解锁操作前完成。锁方式中要进行死锁消除、预防、检测处理。在集中式数据库管理系统中通过两段锁协议可以破坏死锁的形成条件，以保证事务调度的正确性。

但是，在分布式数据库管理系统中，要对各自分散而又有共享要求的数据进行集中统一的管理，并用统一的描述使分散在各地的应用数据在用户看来全局统一在一个数据库中，这给事务的并发控制提出了更高要求。

在分布式环境下，如果在分布式数据库系统中没有重复存储的数据，可以使用分布式锁技术。其实现方法是，对每个节点保留一个局部锁管理器，处理对存储在本地的数据加锁和释放锁的请求。当分布式数据库系统中有多处重复存储的数据时，可采用集中式锁技术，即仅有一个锁管理器，该管理器放在唯一一个节点上，所有的加锁和释放锁的请求都在该节点上处理。此外，在分布式环境下的锁方法，还有混合锁技术、主副本锁协议等。

封锁法实现了一种并发控制，可以满足一般数据库应用对并发控制的要求，但是也有自身的缺点，那就是开销大。事务加锁都有一定的开销，还会降低事务的并发性。以锁为基础的并发控制算法在实际 DDBMS 中应用较为广泛，例如，在 Oracle 系统中并发控制的实现就是采用锁方法。

2. 时间印法

所谓时间印法就是在每个事务启动运行时，并发控制机制赋予其唯一一个时间印（通常为事务的启动时间），以实现多事务的可串行性。后继事务有较大的时间印，事务的时间印一直保持到事务结束。

此外，每个数据实体都有时间印，分为读时间印和写时间印。读时间印即数据上一次被读出的时间，写时间印即数据最近被写入的时间。只有当时间印比数据项上的时间印大或者相当的事务，才能执行对数据的操作并在操作完成后置数据项的时间印为事务和数据项时间印的最大值，否则拒绝令操作重启。这种方法使事务的并行等价于特定的串行序列，即按时间印递增的序列，一般不会产生死锁。

在分布式环境下，如果能够解决所有事务生成唯一时间印的策略，集中式时间印技术就可以直接应用于分布式数据库管理系统。实际上，在分布式策略中，每个节点都使用本地的逻辑计数器或时钟产生局部的时间印，全局的时间印通过在局部的时间印前加上各节点的标识符来获得，这样获得的时间印是唯一的。

3. 唯一时间印

采用时间印的并行控制算法主要有两种：基本时间印法和保守时间印法。主要缺点是使事务重新启动的次数较多。为解决这个问题，提出保守时间印法，该方法的主要特点是不

会拒绝任何操作,从而不会重启事务。

处理方法是当出现冲突操作时,把较新的缓冲起来,等待较旧的操作执行完成后再执行被缓冲起来的操作。这就需要确定何时所有的事务操作均已执行完。但这可能引发死锁的情况,也可能出现某一场地被阻断的情况。事实上,避免死锁的唯一方法是超时后发送一个空操作。

在分布式数据库管理系统中,数据的可用性和可靠性从某种角度看是矛盾的。对于可用性和可靠性要求较高的应用,可以采用封锁和时间印相结合的方法,可以避免错误、避免死锁,必要时要考虑采用容错和重构技术来提高可靠性。

11.3.3 NoSQL 数据库介绍

1. 什么是 NoSQL

大家有没有听说过 NoSQL 呢? 近年,这个词极受关注。看到 NoSQL 这个词,大家可能会误以为是“No! SQL”的缩写,并深感疑惑:“SQL 怎么会没有必要了呢?”但实际上,它是 Not Only SQL 的缩写。它的意义是:适用关系型数据库的时候就使用关系型数据库,不适用的时候也没有必要非使用关系型数据库不可,可以考虑使用更加合适的数据存储。

为弥补关系型数据库的不足,各种各样的 NoSQL 数据库应运而生。

为了更好地了解本书所介绍的 NoSQL 数据库,对关系型数据库的理解是必不可少的。那么,就让我们先来看一看关系型数据库的历史、分类和特征吧。

2. 关系型数据库简史

1969 年,埃德加·弗兰克·科德(Edgar Frank Codd)发表了划时代的论文,首次提出了关系数据模型的概念。但可惜的是,刊登论文的 *IBM Research Report* 只是 IBM 公司的内部刊物,因此论文反响平平。1970 年,他再次在刊物 *Communication of the ACM* 上发表了题为 *A Relational Model of Data for Large Shared Data banks*(大型共享数据库的关系模型)的论文,终于引起了大家的关注。

科德所提出的关系数据模型的概念成为现今关系型数据库的基础。当时的关系型数据库由于硬件性能低劣、处理速度过慢而迟迟没有得到实际应用。但之后随着硬件性能的提升,加之使用简单、性能优越等优点,关系型数据库得到了广泛的应用。

3. 通用性及高性能

虽然这是在讲解 NoSQL 数据库,但有一个重要的大前提,请大家一定不要误解。这个大前提就是“关系型数据库的性能绝对不低,它具有非常好的通用性和非常高的性能”。毫无疑问,对于绝大多数的应用来说它都是最有效的解决方案。

关系型数据库作为应用广泛的通用型数据库,它的突出优势主要有以下几点:

① 保持数据的一致性(事务处理);

② 由于以标准化为前提,数据更新的开销很小(相同的字段基本上都只有一处);

③ 可以进行 JOIN 等复杂查询;

④ 存在很多实际成果和专业技术信息(成熟的技术)。

这其中,能够保持数据的一致性是关系型数据库的最大优势。在需要严格保证数据一致性和处理完整性的情况下,用关系型数据库是肯定没有错的。但是有些情况不需要 JOIN,对上述关系型数据库的优点也没有什么特别需要,这时似乎也就没有必要拘泥于关

系型数据库了。

4. 关系型数据库的不足

就像之前提到的那样,关系型数据库的性能非常高。但是它毕竟是一个通用型的数据库,并不能完全适应所有的用途。具体来说它并不擅长以下处理:

① 大量数据的写入处理;

② 为有数据更新的表做索引或表结构(schema)变更;

③ 字段不固定的应用;

④ 对简单查询需要快速返回结果的处理。

5. NoSQL 数据库弥补关系型数据库的不足

关系型数据库应用广泛,能进行事务处理和 JOIN 等复杂处理。相对地,NoSQL 数据库只应用在特定领域,基本上不进行复杂的处理,但它恰恰弥补了之前所列举的关系型数据库的不足之处。

(1) 易于数据的分散

如前所述,关系型数据库并不擅长大量数据的写入处理。原本关系型数据库就是以 JOIN 为前提的,也就是说,各个数据之间存在关联是关系型数据库得名的主要原因。为了进行 JOIN 处理,关系型数据库不得不把数据存储在同一个服务器内,这不利于数据的分散。相反,NoSQL 数据库原本就不支持 JOIN 处理,各个数据都是独立设计的,很容易把数据分散到多个服务器上。由于数据被分散到了多个服务器上,减少了每个服务器上的数据量,即使要进行大量数据的写入操作,处理起来也更加容易。同理,数据的读入操作当然也同样容易。

(2) 提升性能和增大规模

下面说一点题外话,如果想要使服务器能够轻松地处理更大量的数据,那么只有两个选择:一是提升性能,二是增大规模。下面我们来整理一下这两者的不同。

首先,提升性能指的就是通过提升现行服务器自身的性能来提高处理能力。这是非常简单的方法,程序方面也不需要进行变更,但需要一些费用。若要购买性能翻倍的服务器,需要花费的资金往往不只是原来的 2 倍,可能需要多达 5～10 倍。这种方法虽然简单,但是成本较高。

另一方面,增大规模指的是使用多台廉价的服务器来提高处理能力。它需要对程序进行变更,但由于使用廉价的服务器,可以控制成本。另外,以后只要依葫芦画瓢增加廉价服务器的数量就可以了。

(3) 不对大量数据进行处理是否就没有使用的必要

NoSQL 数据库基本上来说为了"使大量数据的写入处理更加容易(让增加服务器数量更容易)"而设计的。但如果不是对大量数据进行操作,NoSQL 数据库的应用就没有意义吗?

答案是否定的。的确,它在处理大量数据方面很有优势。但实际上 NoSQL 数据库还有以下各种各样的特点,如果能够恰当地利用这些特点,将会是非常有帮助的。

① 希望顺畅地对数据进行缓存(Cache)处理。

② 希望对数组类型的数据进行高速处理。

③ 希望进行全部保存。

6. 多样的 NoSQL 数据库

NoSQL 数据库存在着“key-value 存储”“文档型数据库”“列存储数据库”等各种各样的种类，每种数据库又包含各自的特点。下面让我们一起来了解一下 NoSQL 数据库的种类和特点。

NoSQL 说起来简单，但实际上到底有多少种呢？我在提笔的时候，到 NoSQL 的官方网站上确认了一下，竟然已经有 122 种了。另外官方网站上也介绍了本书没有涉及的图形数据库和对象数据库等各个类别。不知不觉间，已经出现了这么多的 NoSQL 数据库。

下面将为大家介绍具有代表性的 NoSQL 数据库。

1）key-value 存储

这是最常见的 NoSQL 数据库，它的数据是以 key-value 的形式存储的。虽然它的处理速度非常快，但是基本上只能通过 key 的完全一致查询获取数据。根据数据的保存方式可以分为临时性、永久性和两者兼具三种。

（1）保存方式为临时性

memcached 属于临时性类型。所谓临时性就是“数据有可能丢失”的意思。memcached 把所有数据都保存在内存中，这样保存和读取的速度非常快，但是当 memcached 停止时，数据就不存在了。由于数据保存在内存中，所以无法操作超出内存容量的数据(旧数据会丢失)。

① 在内存中保存数据；

② 可以进行非常快速的保存和读取处理；

③ 数据有可能丢失。

（2）保存方式为永久性

Tokyo Tyrant、Flare、ROMA 等属于永久性类型。和临时性相反，所谓永久性就是“数据不会丢失”的意思。这里的 key-value 存储不像 memcached 那样在内存中保存数据，而是把数据保存在硬盘上。与 memcached 在内存中处理数据比起来，由于必然要发生对硬盘的 I/O 操作，所以性能上还是有差距的。但数据不会丢失是它最大的优势。

① 在硬盘上保存数据；

② 可以进行非常快速的保存和读取处理(但无法与 memcached 相比)；

③ 数据不会丢失。

（3）保存方式兼具临时性和永久性

Redis 属于两者兼具。Redis 有些特殊，兼具临时性和永久性，且集合了临时性 key-value 存储和永久性 key-value 存储的优点。Redis 首先把数据保存到内存中，在满足特定条件(默认是 15 分钟一次以上，5 分钟内 10 个以上，1 分钟内 10 000 个以上的 key 发生变更)的时候将数据写入到硬盘中。这样既确保了内存中数据的处理速度，又可以通过写入硬盘来保证数据的永久性。这种类型的数据库特别适合于处理数组类型的数据。

① 同时在内存和硬盘上保存数据；

② 可以进行非常快速的保存和读取处理；

③ 保存在硬盘上的数据不会消失(可以恢复)；

④ 适合于处理数组类型的数据。

2）面向文档的数据库

MongoDB、CouchDB 属于这种类型。它们属于 NoSQL 数据库，但与 key-value 存储

相异。

① 不定义表结构。面向文档的数据库具有以下特征：即使不定义表结构，也可以像定义了表结构一样使用。关系型数据库在变更表结构时比较费事，而且为了保持一致性还需修改程序。然而 NoSQL 数据库则可省去这些麻烦（通常程序都是正确的），确实是方便快捷。

② 可以使用复杂的查询条件。跟 key-value 存储不同的是，面向文档的数据库可以通过复杂的查询条件来获取数据。虽然不具备事务处理和 JOIN 这些关系型数据库所具有的处理能力，但除此以外的其他处理基本上都能实现。这是非常容易使用的 NoSQL 数据库。

③ 不需要定义表结构。

④ 可以利用复杂的查询条件。

3）面向列的数据库

Cassandra、Hbase、HyperTable 属于这种类型。由于近年来数据量出现爆发性增长，这种类型的 NoSQL 数据库尤其引人注目。面向列的数据库是以列为单位来存储数据的，擅长以列为单位读入数据。

面向列的数据库具有高扩展性，即使数据增加也不会降低相应的处理速度（特别是写入速度），所以它主要应用于需要处理大量数据的情况。另外，利用面向列的数据库的优势，把它作为批处理程序的存储器来对大量数据进行更新也是非常有用的。但由于面向列的数据库跟现行数据库存储的思维方式有很大不同，应用起来十分困难。

4）面向行的数据库

普通的关系型数据库都是以行为单位来存储数据的，擅长进行以行为单位的读入处理，比如特定条件数据的获取。因此，关系型数据库也被称为面向行的数据库。

11.3.4　HBase 介绍

HBase 是一个面向列的分布式数据库。HBase 不是一个关系型数据库，其设计目标是用来解决关系型数据库在处理海量数据时的理论和实现上的局限性。传统关系型数据库在 20 世纪 70 年代为交易系统设计，以满足数据一致性（ACID）为目标，并没有考虑数据规模扩大时的扩展性，以及单点系统失效时的可靠性。虽然经过多年的技术发展产生了一些对关系型数据库的修补（并行数据库），然而受限于理论和实现上的约束，扩展性从来没有超过 40 个服务器节点。而 HBase 从一开始就是为 Terabyte 到 Petabyte 级别的海量数据存储和高速读写而设计，这些数据要求能够被分布在数千台普通服务器上，并且能够被大量并发用户高速访问。

HBase 从 2008 年第一次商用开始，已经被越来越多的在线服务公司所采用。其中最大的是 Facebook 新上线的整合了 E-mail、SNS、Chat 和短消息的在线即时消息系统。

1. 高可扩展性

HBase 是真正意义上的线性水平扩展。数据量累积到一定程度（可配置），HBase 系统会自动对数据进行水平切分，并分配不同的服务器来管理这些数据。这些数据可以扩散到上千个普通服务器上。这样一方面可以由大量普通服务器组成大规模集群，来存放海量数据（从几个 TB 到几十 PB 的数据）。另一方面，当数据峰值接近系统设计容量时，可以简单通过增加服务器的方式来扩大容量。这个动态扩容过程无须停机，HBase 系统可以照常运行并提供读写服务，完全实现动态无缝无宕机扩容。

2. 高性能

HBase 的设计目的之一是支持高并发用户数的高速读写访问。这是通过两方面来实现的。首先数据行被水平切分并分布到多台服务器上，在大量用户访问时，访问请求也被分散到了不同的服务器上，虽然每个服务器的服务能力有限，但是数千台服务器汇总后可以提供极高性能的访问能力。其次，HBase 设计了高效的缓存机制，有效提高了访问的命中率，提高了访问性能。

3. 高可用性

HBase 建立在 HDFS 之上。HDFS 提供了数据自动复制和容错的功能。HBase 的日志和数据都存放在 HDFS 上，即使在读写过程中当前服务器出现故障(硬盘、内存、网络等故障)，日志也不会丢失，数据都可以从日志中自动恢复。HBase 系统会自动分配其他服务器接管并恢复这些数据。因此一旦成功写入数据，这些数据就保证被持久化并被冗余复制，整个系统的高可用性得到保证。

4. 数据模型及其特点

HBase 是一个面向列的、稀疏的、分布式的、持久化存储的多维排序映射表(Map)。表的索引是行关键字、列簇名(Column Family)、列关键字以及时间戳；表中的每个值都是一个未经解析的字节数组。

面向列：指的是同一个列簇里所有数据都存放在一个文件中，从而在读写时有效降低磁盘 I/O 的开销，并且由于类似数据存放在一起，提高了压缩比。经过压缩后的数据容量通常达到原来的 1/3 到 1/5，极大节省了存储空间。

多维表：这是对传统二维关系表的极大扩充。传统二维表有两维：行和列。列在设计表结构时必须预先固定，而行可以动态增加，也就是说有一个维度可动态改变。HBase 的多维表有四维，列簇需要在设计表结构时事先确定，而行、列、时间维都可以动态增加。也就是说有三个维度可动态改变。这种结构非常适合用来表述有嵌套关系的数据。另外，动态增删列的能力也给很多业务带来便利，特别是这些业务在不停地演化，需要的列字段也在不停地增加，多维表结构可以随时进行改变以适应业务发展需求。

稀疏表：由于多维表的列可以动态增加，必然导致不同行相同列的数据大部分为空，也就是说这个表是稀疏的。不像传统关系型数据库，HBase 不存放空值，只存放有内容的表格单元(cell)，因此可以支持超大稀疏表，而不会带来任何开销。这对传统的表结构设计也带来了观念上的大改变。

HBase 是 Google Bigtable 的开源实现，类似 Google Bigtable 利用 GFS 作为其文件存储系统，HBase 利用 Hadoop HDFS 作为其文件存储系统；Google 运行 MapReduce 来处理 Bigtable 中的海量数据，HBase 同样利用 Hadoop MapReduce 来处理 HBase 中的海量数据；Google Bigtable 利用 Chubby 作为协同服务，HBase 利用 Zookeeper 作为对应。如图 11-3 所示。

图 11-3 描述了 Hadoop EcoSystem 中的各层系统，其中 HBase 位于结构化存储层，Hadoop HDFS 为 HBase 提供了高可靠性的底层存储支持，Hadoop MapReduce 为 HBase 提供了高性能的计算能力，Zookeeper 为 HBase 提供了稳定服务和 failover 机制。

此外，Pig 和 Hive 还为 HBase 提供了高层语言支持，使得在 HBase 上进行数据统计处理变得非常简单。Sqoop 则为 HBase 提供了方便的 RDBMS 数据导入功能，使得传统数据

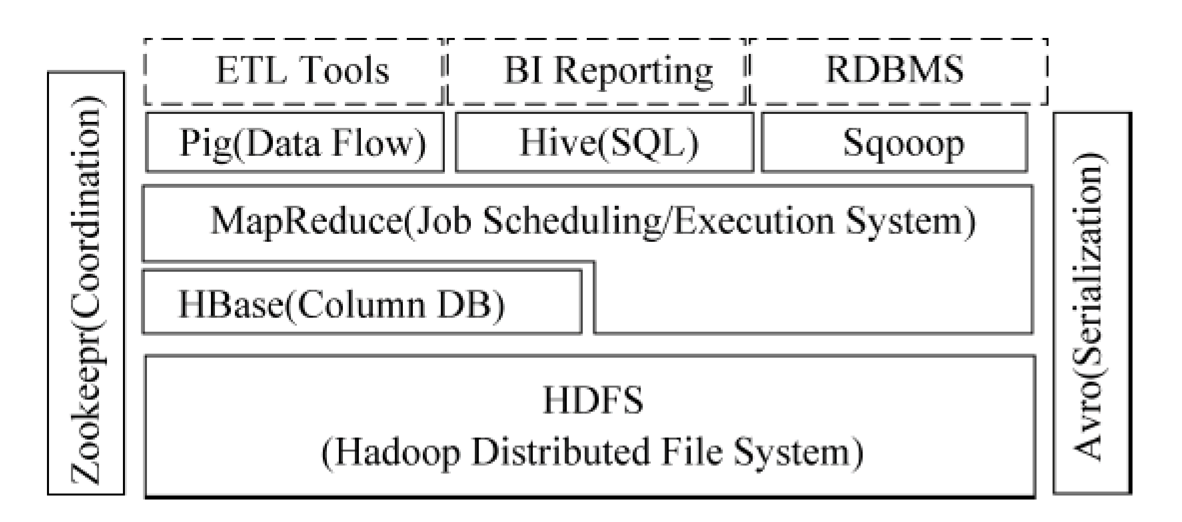


图 11-3　各层系统结构

库数据向 HBase 中迁移变得非常方便。

5. HBase 访问接口

(1) Native Java API：这是最常规和高效的访问方式，适合 Hadoop MapReduce Job 并行批处理 HBase 表数据。

(2) HBase Shell：HBase 的命令行工具，有最简单的接口，适合 HBase 管理使用。

(3) Thrift Gateway：利用 Thrift 序列化技术，支持 C++、PHP、Python 等多种语言，适合其他异构系统在线访问 HBase 表数据。

(4) REST Gateway：支持 REST 风格的 HTTP API 访问 HBase，解除了语言限制。

(5) Pig：可以使用 Pig Latin 流式编程语言来操作 HBase 中的数据。与 Hive 类似，本质最终也是编译成 MapReduce Job 来处理 HBase 表数据，适合做数据统计。

(6) Hive：当前 Hive 的 Release 版本尚没有加入对 HBase 的支持，但在下一个版本 Hive 0.7.0 中将会支持 HBase，可以使用 SQL 语言等来访问 HBase。

6. Table 和 Region

当 Table 随着记录数不断增加而变大后，会逐渐分裂成多份 splits，成为 regions。一个 region 由 startkey 或 endkey 表示，不同的 region 会被 Master 分配给相应的 RegionServer 进行管理，如图 11-4 所示。

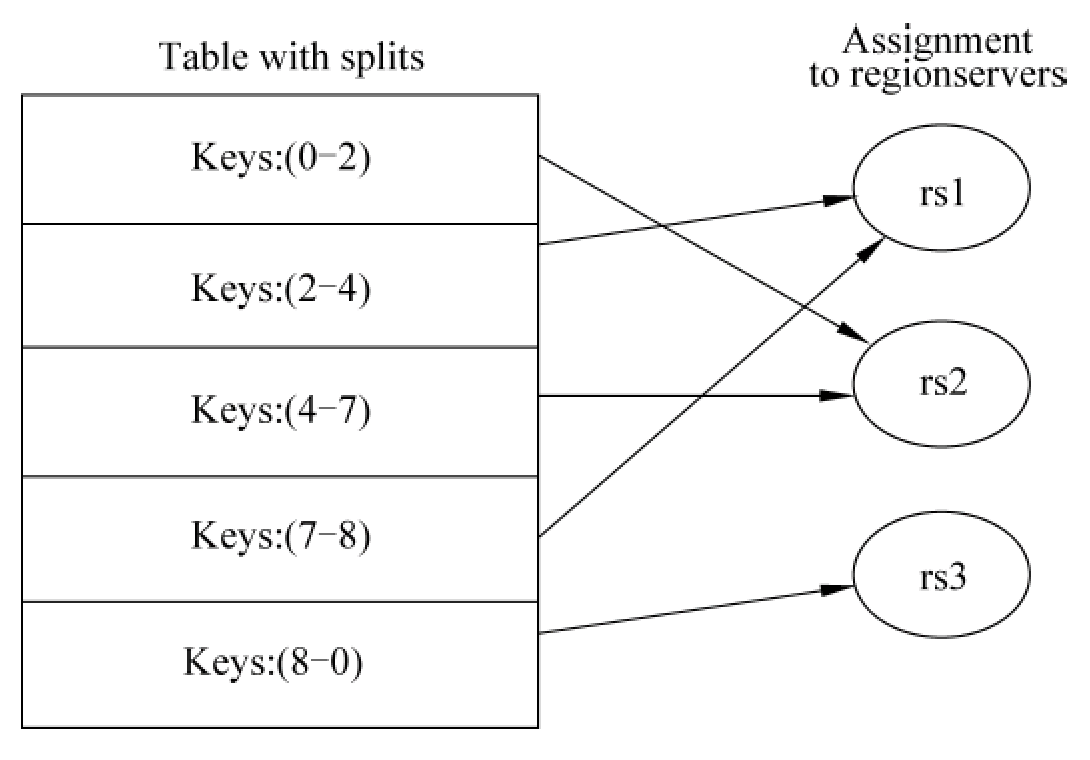


图 11-4　分配管理图

HBase 中有两张特殊的 Table，即“.META.”和“-ROOT-”。

∅ .META.：记录了用户表的 Region 信息，“.META.”可以有多个 regoin。

∅ -ROOT-：记录了“.META.”表的 Region 信息，“-ROOT-”只有一个 region。

ø Zookeeper 中记录了“－ROOT－”表的 location。

客户端访问用户数据之前需要首先访问 zookeeper，然后访问“－ROOT－”表，接着访问“. META. ”表，最后才能找到用户数据的位置去访问，中间需要多次网络操作，不过客户端会做 cache 缓存。

7．MapReduce on HBase

在 HBase 系统上运行批处理运算，最方便和实用的模型依然是 MapReduce。

HBase Table 和 Region 的关系，比较类似于 HDFS File 和 Block 的关系，HBase 提供了配套的 TableInputFormat 和 TableOutputFormat API，可以方便地将 HBase Table 作为 Hadoop MapReduce 的 Source 和 Sink，对于 MapReduce Job 应用开发人员来说，基本不需要关注 HBase 系统自身的细节。

8．Client

HBase Client 使用 HBase 的 RPC 机制与 HMaster 和 HRegionServer 进行通信，对于管理类操作，Client 与 HMaster 进行 RPC；对于数据读写类操作，Client 与 HRegionServer 进行 RPC。

9．Zookeeper

Zookeeper Quorum 中除了存储“－ROOT－”表的地址和“HMaster”的地址，HRegionServer 也会把自己以 Ephemeral 方式注册到 Zookeeper 中，使得 HMaster 可以随时感知到各个 HRegionServer 的健康状态。此外，Zookeeper 也避免了 HMaster 的单点问题，见下文描述。

10．HMaster

HMaster 没有单点问题，HBase 中可以启动多个 HMaster，通过 Zookeeper 的 Master Election 机制保证总有一个 Master 运行，HMaster 在功能上主要负责 Table 和 Region 的以下管理工作。

（1）管理用户对 Table 的增、删、改、查操作。

（2）管理 HRegionServer 的负载均衡，调整 Region 分布。

（3）在 Region Split 后，负责新 Region 的分配。

（4）在 HRegionServer 停机后，负责失效 HRegionServer 上的 Regions 迁移。

11．HRegionServer

HRegionServer 主要负责响应用户 I/O 请求，向 HDFS 文件系统中读写数据，是 HBase 中最核心的模块，如图 11-5 所示。

HRegionServer 内部管理了一系列 HRegion 对象，每个 HRegion 对应着 Table 中的一个 Region，HRegion 由多个 HStore 组成。每个 HStore 对应了 Table 中的一个 Column Family 的存储，可以看出每个 Column Family 其实就是一个集中的存储单元，因此最好将具备共同 I/O 特性的 column 放在一个 Column Family 中，这样最高效。

HStore 存储是 HBase 存储的核心了，其中由两部分组成，一部分是 MemStore，一部分是 StoreFiles。MemStore 是 Sorted Memory Buffer，用户写入的数据首先会放入 MemStore，当 MemStore 满了以后会刷新成一个 StoreFile（底层实现是 HFile）。当 StoreFile 文件数量增长到一定阈值，会触发 Compact 合并操作，将多个 StoreFiles 合并成一个 StoreFile，合并过程中会进行版本合并和数据删除，因此可以看出 HBase 其实只有增

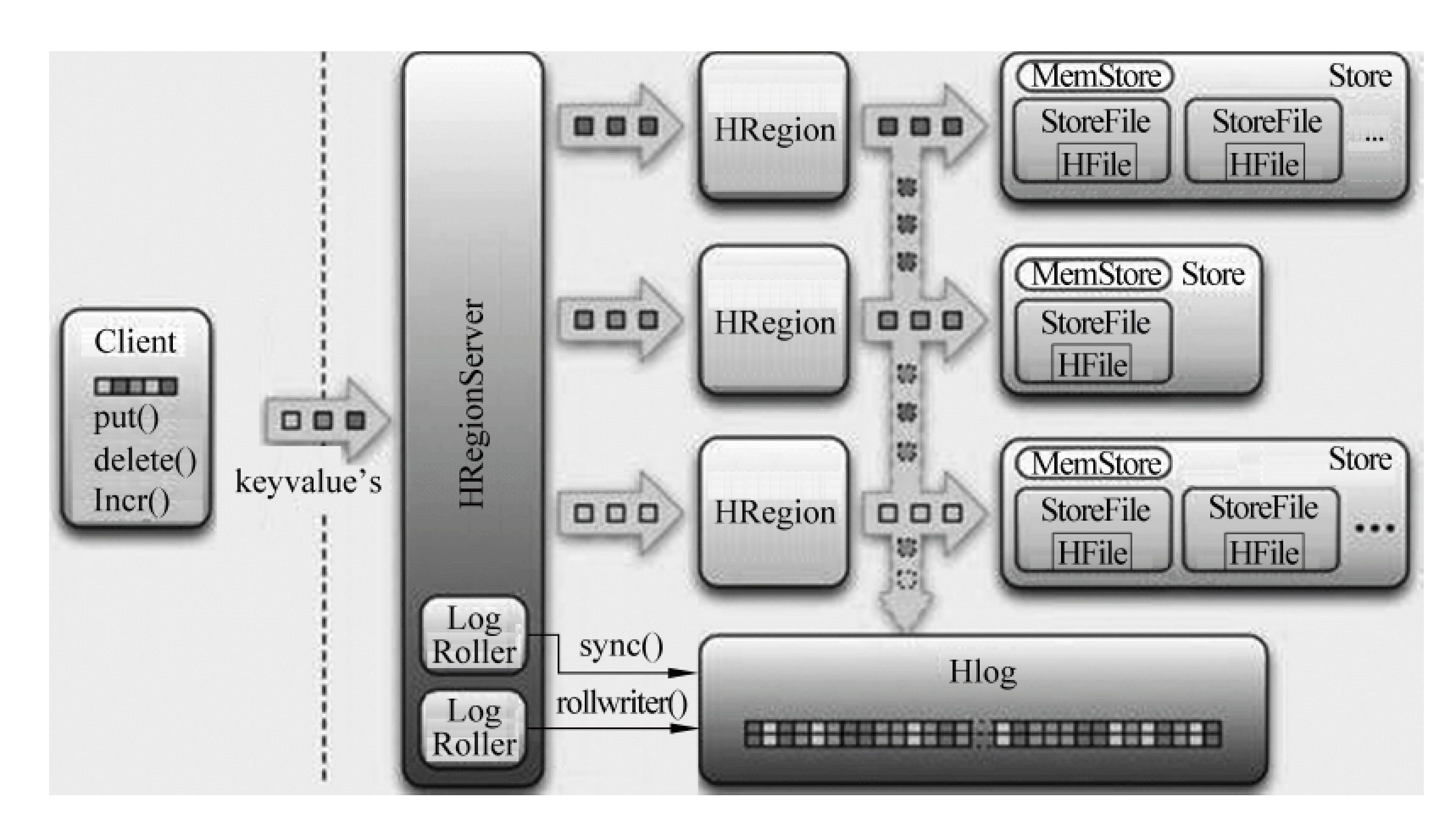


图 11-5　HRegionServe 结构示意图

加数据，所有的更新和删除操作都是在后续的 compact 过程中进行的，这使得用户的写操作只要进入内存中就可以立即返回，保证了 HBase I/O 的高性能。当 StoreFiles 合并后，会逐步形成越来越大的 StoreFile，当单个 StoreFile 大小超过一定阈值后，会触发 Split 操作，同时把当前 Region 分成两个，父 Region 会下线，新分出的两个孩子 Region 会被 HMaster 分配到相应的 HRegionServer 上，使得原先一个 Region 的压力得以分流到两个 Region 上。

在理解了上述 HStore 的基本原理后，还必须了解一下 HLog 的功能，因为上述的 HStore 在系统正常工作的前提下是没有问题的，但是在分布式系统环境中，无法避免系统出错或者宕机，因此一旦 HRegionServer 意外退出，MemStore 中的内存数据将会丢失，这就需要引入 HLog 了。每个 HRegionServer 中都有一个 HLog 对象，HLog 是一个实现 Write Ahead Log 的类，在每次用户操作写入 MemStore 的同时，也会写一份数据到 HLog 文件中（HLog 文件格式见后续），HLog 文件定期会滚动出新的，并删除旧的文件（已持久化到 StoreFile 中的数据）。当 HRegionServer 意外终止后，HMaster 会通过 Zookeeper 感知到，HMaster 首先会处理遗留的 HLog 文件，将其中不同 Region 的 Log 数据进行拆分，分别放到相应 region 的目录下，然后再将失效的 region 重新分配，领取到这些 region 的 HRegionServer 在 Load Region 的过程中会发现有历史 HLog 需要处理，因此会转发 HLog 中的数据到 MemStore 中，然后刷新到 StoreFiles 中，完成数据恢复。

12. HBase 存储格式

HBase 中的所有数据文件都存储在 Hadoop HDFS 文件系统上，主要包括上述提出的两种文件类型。

(1) HFile：HBase 中 Key Value 数据的存储格式，HFile 是 Hadoop 的二进制格式文件，实际上 StoreFile 就是对 HFile 做了轻量级包装，即 StoreFile 底层就是 HFile。

(2) HLog File：HBase 中 WAL(Write Ahead Log)的存储格式，物理上是 Hadoop 的 Sequence File。

13. HFile

图 11-6 是 HFile 的存储格式。

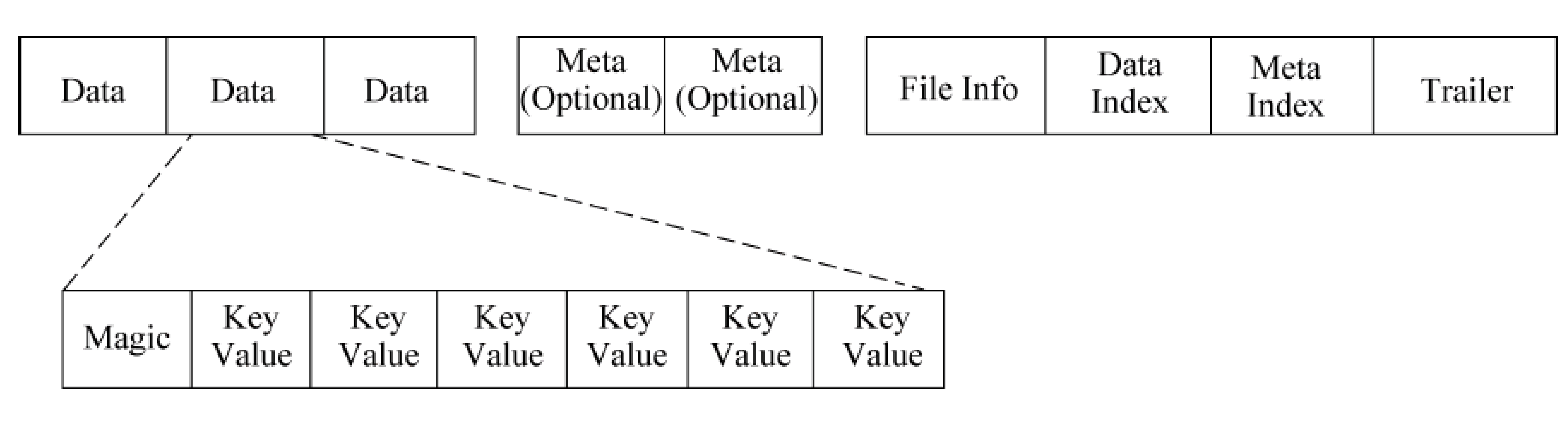


图 11-6　HFile 的存储格式

首先 HFile 文件是不定长的，长度固定的只有其中的两块：Trailer 和 FileInfo。正如图中所示，Trailer 中有指针指向其他数据块的起始点。File Info 中记录了文件的一些 Meta 信息，例如：AVG_KEY_LEN、AVG_VALUE_LEN、LAST_KEY、COMPARATOR、MAX_SEQ_ID_KEY 等。Data Index 和 Meta Index 块记录了每个 Data 块和 Meta 块的起始点。

Data Block 是 HBase I/O 的基本单元，为了提高效率，HRegionServer 中有基于 LRU 的 Block Cache 机制。每个 Data 块的大小可以在创建一个 Table 的时候通过参数指定，大号的 Block 有利于顺序扫描，小号 Block 利于随机查询。每个 Data 块除了开头的 Magic 以外就是一个个 KeyValue 对拼接而成，Magic 内容就是一些随机数字，目的是防止数据损坏。后面会详细介绍每个 KeyValue 对的内部构造。

HFile 里面的每个 KeyValue 对就是一个简单的 byte 数组。但是这个 byte 数组里面包含了很多项，并且有固定的结构。我们来看看里面的具体结构，如图 11-7 所示。

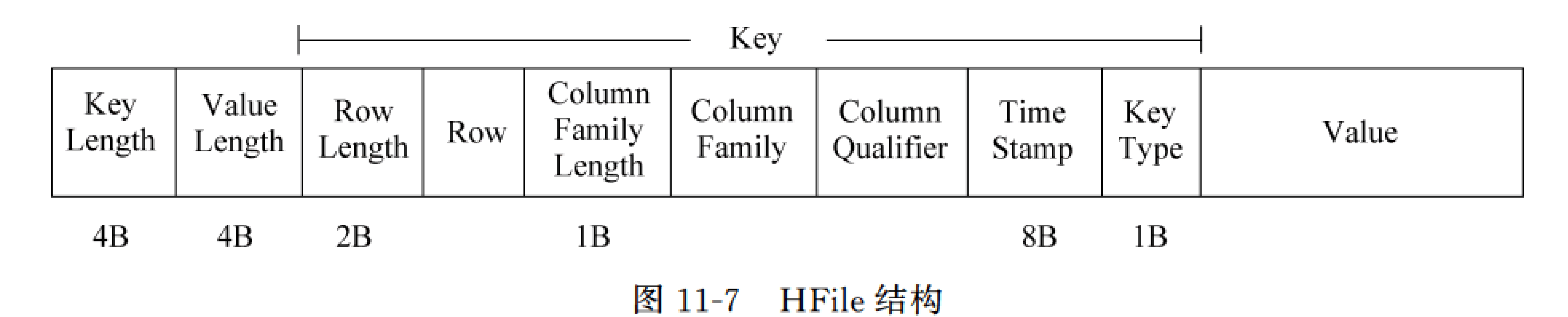


图 11-7　HFile 结构

开始是两个固定长度的数值，分别表示 Key 的长度和 Value 的长度。紧接着是 Key，开始是固定长度的数值，表示 RowKey 的长度，紧接着是 RowKey。然后是固定长度的数值，表示 Family 的长度。接下来是 Family，接着是 Qualifier。最后是两个固定长度的数值，表示 Time Stamp 和 Key Type(Put/Delete)。Value 部分没有这么复杂的结构，就是纯粹的二进制数据。

14. HLogFile

图 11-8 中示意了 HLog 文件的结构，其实 HLog 文件就是一个普通的 Hadoop Sequence File，Sequence File 的 Key 是 HLogKey 对象，HLogKey 中记录了写入数据的归属信息，除了 table 和 region 名字外，同时还包括 sequence number 和 timestamp，timestamp 是"写入时间"，sequence number 的起始值为 0，或者是最近一次存入文件系统中的 sequence number。

HLogSequece File 的值(Value)是 HBase 的 KeyValue 对象，即对应 HFile 中的 KeyValue。

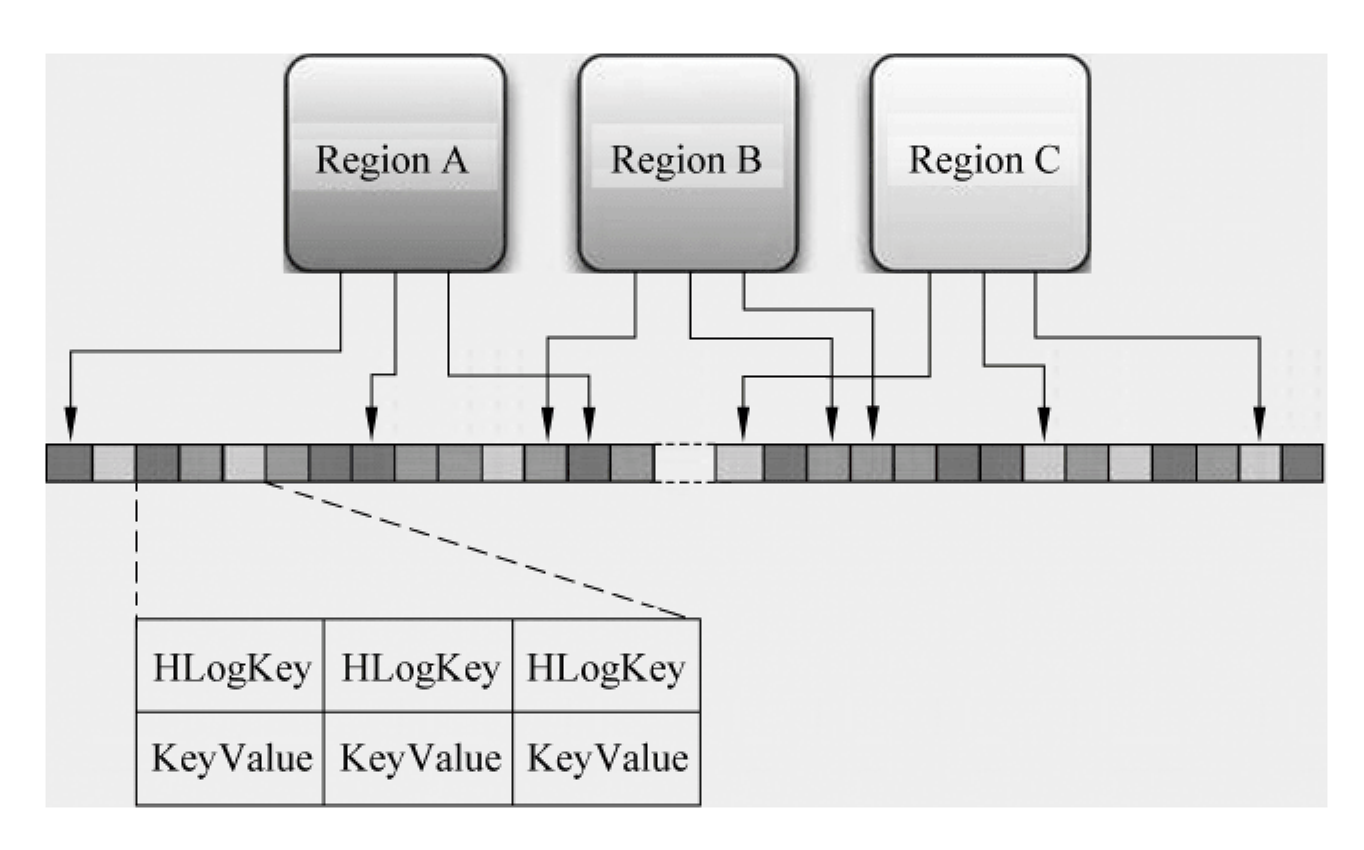


图 11-8 HFile 文件的结构

11.4 关键技术分析

前面从系统的角度介绍了一些典型的分布式存储系统，其中都需要解决分布式环境中的一致性与复制、负载均衡、容错等问题。但是，还有一些关键技术可以改善存储系统的性能。

11.4.1 元数据管理

元数据是为了简化检索、使用和管理信息资源的结构化数据。在大规模对象存储系统以及分布式文件系统中，通常会将元数据管理和文件读取分离，比如 GFS 中的 Master 节点，虽然相对于文件来说，元数据很小，但是元数据操作占整个文件系统操作的 50%，所以元数据服务器集群的行为直接影响整个系统的性能和可扩展性。MDS(Meta Data Server)集群就是要采取某种分区策略，将元数据分布到多个节点上，实现元数据的高效访问和集群的负载均衡。

静态子树分割(sub-tree partitioning)需要系统管理员来确定文件系统如何分布，从而给 MDS 分配相应的目录层次子树，这种方法允许存储系统横向扩展，但不能纵向扩展，基于传统网络文件系统的系统采用这种方法。基于哈希的方法是根据文件路径名或者其他唯一标识符来映射元数据所在位置，客户端可以直接定位和联系 MDS，如果所采用的哈希函数设计得高效均匀，用户请求就会在集群中均衡分布，但是哈希策略消除了文件的层次结构概念，所以丧失了局部性。动态子树分割将文件系统目录层次结构中的不同子树分布到不同的元数据服务器中，分割的粒度更小，管理灵活，能够根据元数据服务器的负载情况实现动态的负载均衡，能够适应动态改变的文件系统和工作负载，缺点是层次遍历过程开销很大。语义感知的元数据管理方法通过把相关文件的元数据聚集到一个组中，从而有利于构建高效的缓存，缩小文件查询的范围，提高系统的可扩展性。

面对大数据环境，一个主控节点用来管理系统中的所有元数据在高并发访问的情况下势必会成为系统的瓶颈，通过客户端的预取和缓存技术只是在一定程度上改善了这种情况，分布式存储系统中元数据管理仍然是一个研究热点。

11.4.2 数据去重

在各个领域中,许多用户数据中含有大量相同或相似的数据,比如用户可能上传大量相同的图片,在云存储领域中多个用户可能共享同一个物理存储,重复或者相似数据出现的概率很高,所以可以利用重复数据删除技术来减少存储系统的开销,提高缓存利用率。与传统的压缩技术不同,去重的目的是消除大数据集中文件内部和文件之间的冗余。重复数据删除技术广泛应用在备份系统中,分为源端、目的端和网内重删,以及归档、RAM、SSD 等场景中,不同的存储环境中去重系统需要根据特定的延迟和吞吐量等需求具体设计。数据去重的粒度又分为文件级和块级,根据去重范围的不同,分为局部和全局去重。在文件写入的 I/O 路径中进行对文件块的指纹比对,如果匹配成功,则直接构建索引指向已有的数据块。去重的方法很多,关键点在于如何高效查找重复的段,如何构建高效索引技术。

在大数据存储环境下,将重复数据删除技术部署在分布式存储系统中,从而实现冗余数据的在线去重,提高存储性能和存储利用率。在重删集群中,关键技术是根据具体的负载进行数据分析和处理,设计智能的数据路由机制,将相似的文件尽可能路由到同一个节点上进行去重,保证节点间负载均衡,提高集群的重删收益,同时确保存储系统整体的性能和带宽。

11.4.3 数据分布和负载均衡

与单机存储系统不同,在分布式存储系统中数据分布在多个节点上,并且为了保证高可用性,会采用多副本的策略将数据存储多份,所以就需要一定的策略,将数据分布在系统的多个节点上,同时保证多个节点的负载均衡。常见的分布算法是哈希分布和顺序分布。传统的哈希分布是根据数据的某个特征值和选定的哈希函数计算哈希值,建立起和存储节点的映射关系,所以该算法的关键在于找到一个散列特性很好的哈希函数,而且在节点个数发生变化时,Rehash 会导致大量数据的迁移。关于一致性哈希有了很大的改进,在节点加入和删除时只会影响到哈希环中的相邻节点,比如上述讲到的 Amazon Dynamo 系统使用的就是改进的一致性哈希算法。顺序分布用在分布式表格存储系统中,将表示大量结构化数据的一个大表顺序划分为多个子表,而后分配到多个存储节点上,随着数据的插入和删除,每个节点上的子表会变得大小不一、不均匀,所以要考虑子表的合并和拆分问题,以及给系统带来的性能影响。相比而言,哈希分布针对的是随机读写操作,不支持顺序和范围扫描,顺序分布更能体现数据的局部性原理,利用预取来改善性能,当然,这和面对的应用场景的数据负载有关。

数据分布算法还需要考虑到各个节点之间的负载均衡,影响负载的因素有很多:新的节点加入、某个节点宕机、CPU、内存、网络带宽等,当检测到不均衡发生时就要数据迁移,以及主副本之间的切换,迁移过程对用户透明。负载均衡算法的研究相对比较成熟,主要有静态和动态负载均衡算法。

在分布式存储系统中负载均衡的实施主要有集中控制和分布式方式。集中的控制方式一般由主节点来实现,简单,通信开销小,但是存在单点故障问题;而分布式的负载均衡算法则不存在单点故障问题,易于扩展,但是当系统规模很大的时候会带来巨大的通信开销,影响系统的整体性能。在大数据的应用如 Hadoop 中有很多研究致力于改进其中的负载均衡算法。

11.5　不同数据库公司的大数据主张

目前全球大数据企业主要分为两大阵营。一部分属于单纯以大数据技术为核心的创新型公司，希望为市场带来创新方案并推动技术发展。另有一些则是以数据库/数据仓储业务为主的知名公司，利用自身资源与技术优势地位冲击大数据领域。

下面介绍几个典型公司的大数据主张。

1. IBM

IBM 在软件领域的地位无须多说，从大数据产品方面来讲，IBM 提供了丰富的产品，从硬件到基于云计算的基础架构，到数据库、数据仓库、大数据平台和面向各个领域的挖掘分析产品。其中，DB2、Informix 与 InfoSphere 数据库平台、Cognos 与 SPSS 分析应用可谓最为知名。IBM 同时也为 Hadoop 开源数据分析平台提供支持。

IBM 的大数据平台是 Hadoop 发行版 BigInsights，如图 11-9 所示。

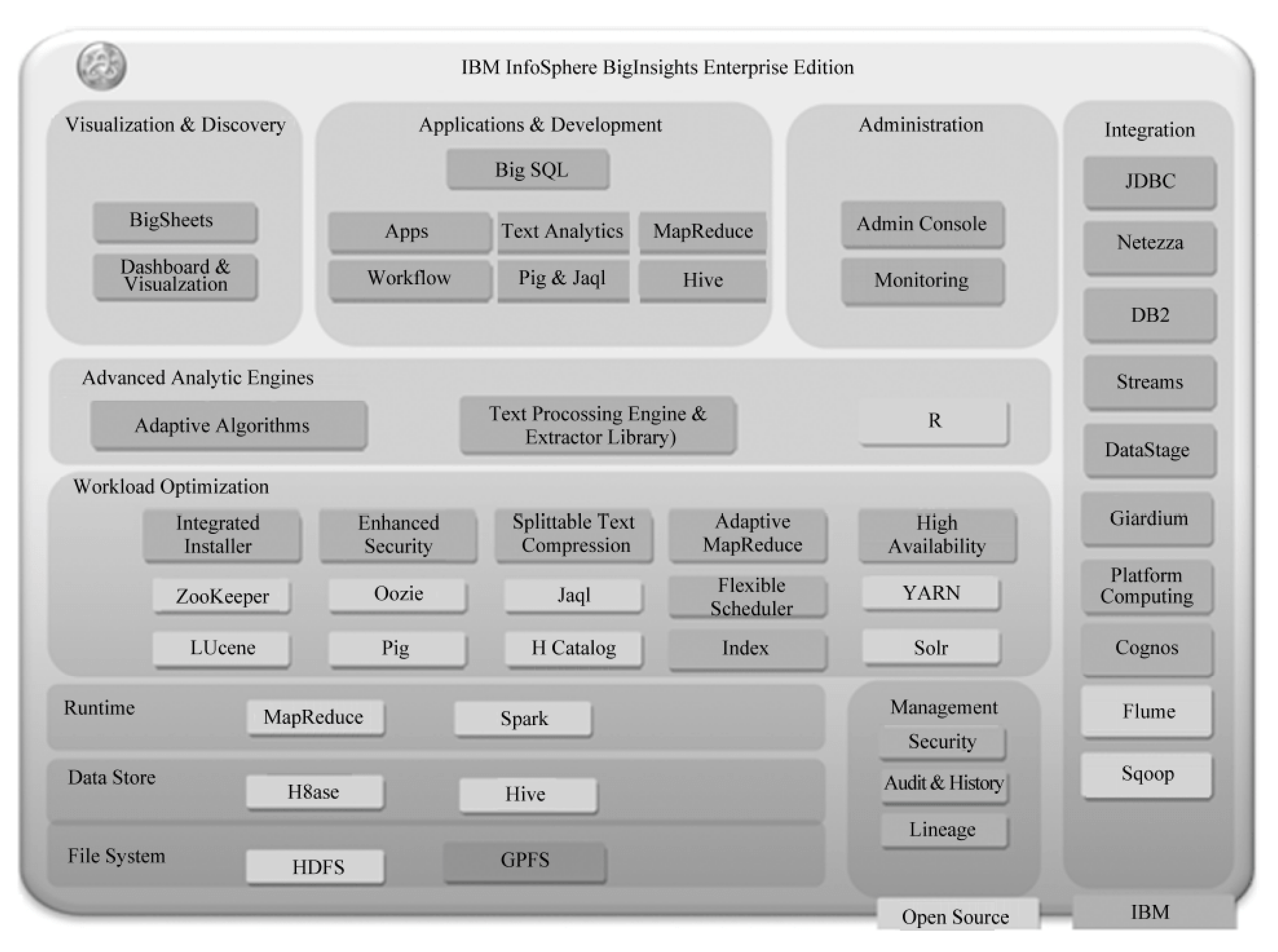


图 11-9　BigInsights 系统架构图

关于技术团队，IBM 也是投入了大量研发力量，一方面，自身投入大量研发团队对开源系统进行研发；另一方面，投入大量资金收购多家公司。

IBM 以其丰富的产品和技术，广泛的客户基础，连年坐拥大数据市场第一宝座，在大数据与分析领域，IBM 不断开拓新的市场，正在帮助包括能源、医疗、金融服务、零售和快消等不同领域的企业实现转型。

2. EMC

易安信为一家美国信息存储资讯科技公司，主要业务为信息存储及管理产品、服务和解决方案。EMC 一方面帮助客户保存并分析大数据，另外也充当着大数据分析智囊“营销科学实验室”的所在地——这家实验室专门分析营销类数据。EMC 推出的最新爆炸性消息是与 VMware 及通用电气一道支持 Pivotal 公司。Pivotal 将对 Hadoop 与 EMC 的 Greenplum 数据库与 HAWQ 查询工具进行整合。EMC 的主要产品为企业级服务器存储硬件和软件，以及与存储相关的网络产品。

EMC 提供了系列产品支持大数据业务，包括 Greenplum(MPP 集群)、PivotalHD、HAWQ(SQL on Hadoop)、Gemfire(内存数据库)，其中 PivotalHD 是 EMC 的 Hadoop 发行版。根据最新消息，EMC 的 Greenplum、HAWQ、Gemfire 已经开源，而 EMC 公司已经被 Dell 并购。如图 11-10 所示。

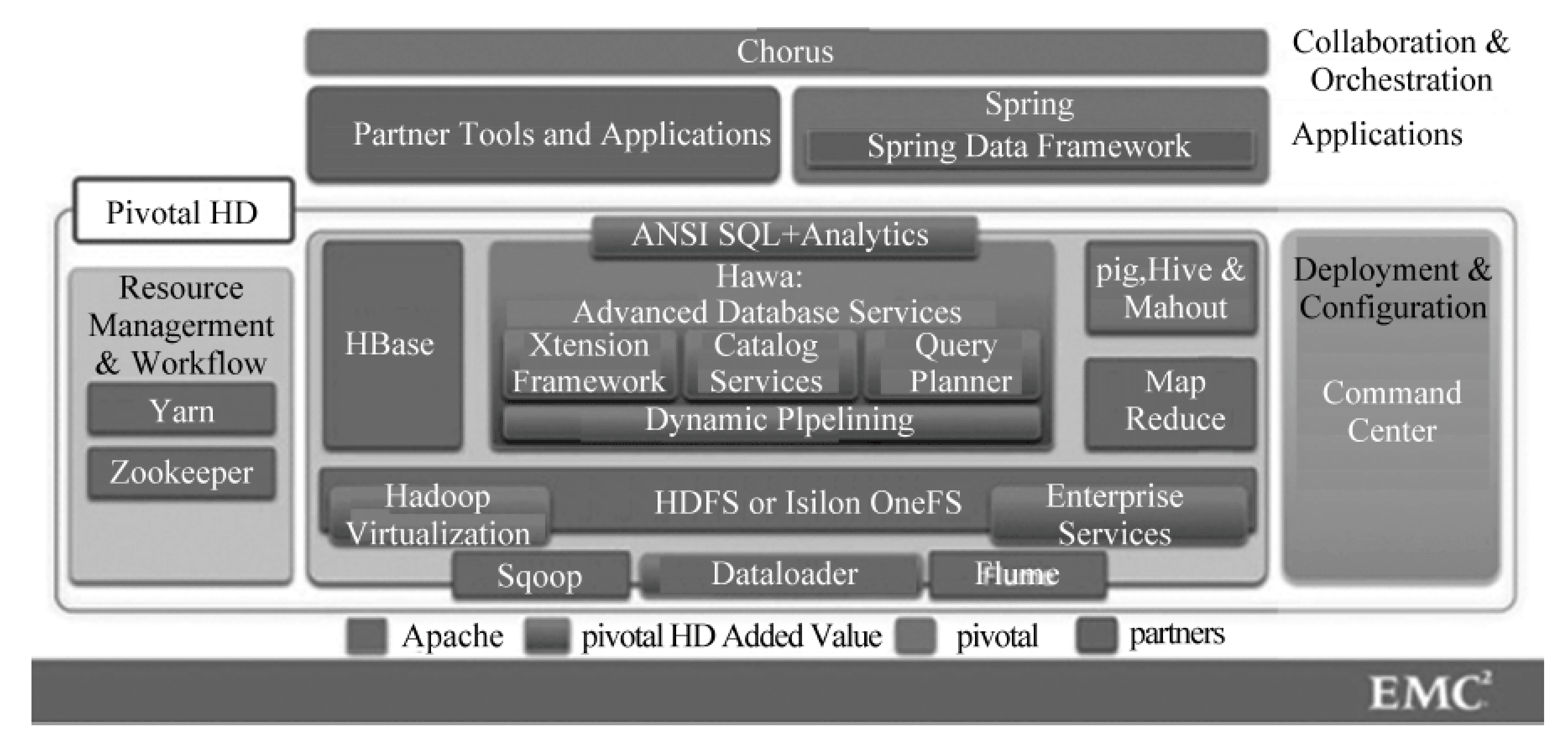


图 11-10 Pivotal 架构图

3. Cloudera

其创建于 2008 年，利用 Hadoop 这一开源技术帮助公司构建大数据平台，为企业等组织型客户提供大数据服务的基础，也做解决方案咨询和认证培训。在 Hadoop 生态领域，Cloudera 是规模最大、知名度最高的企业，也是当前大数据领域最强有力的解决方案服务商之一。

公司提供基于 Hadoop 的单一产品，CDH 是 Cloudera 的 Hadoop 发行版。

CDH 企业版包括分布式存储、数据库、SQL 引擎、机器学习引擎、全文检索、实时流处理、分布式数据总线服务、ETL 工作流引擎、内存计算引擎、元数据生命周期管理、统一多层次安全模块、平台管理、云平台自动部署等企业级 Hadoop 软件模块。如图 11-11 所示。

4. 人大金仓

其创建于 1999 年，是中国自主可控数据库、数据管理全线产品及解决方案的领军企业。通过自主研发、产学研结合、兼并收购等方式，快速形成了集数据存储、管理、集成、分析、展现于一体的全面产品线及符合新一代架构的数据中心解决方案。产品包括数据采集 KingbaseDI、数据存储、DBCloud、数据分析以及可视化 KingbaseSmartBI 等。如图 11-12 所示。

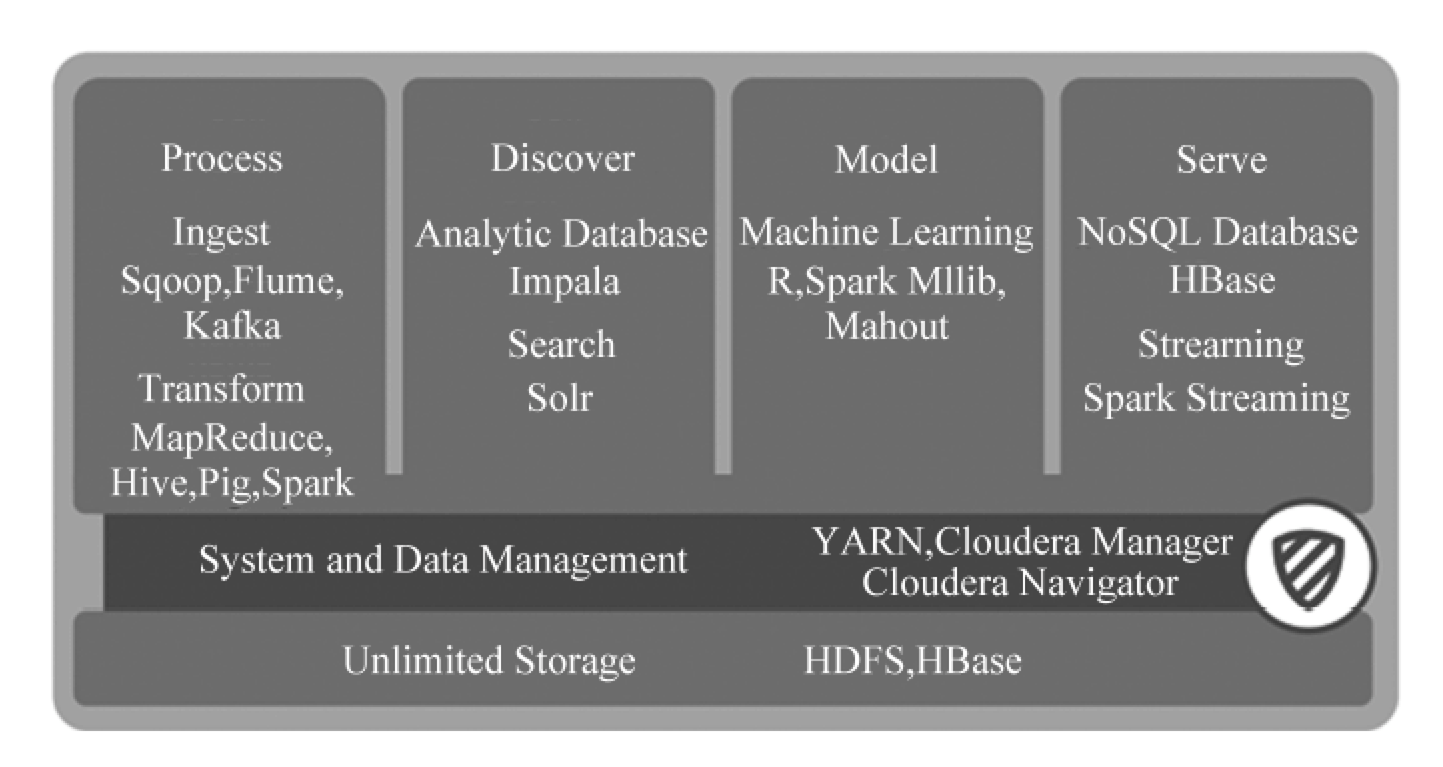


图 11-11　Hadoop 系统架构图

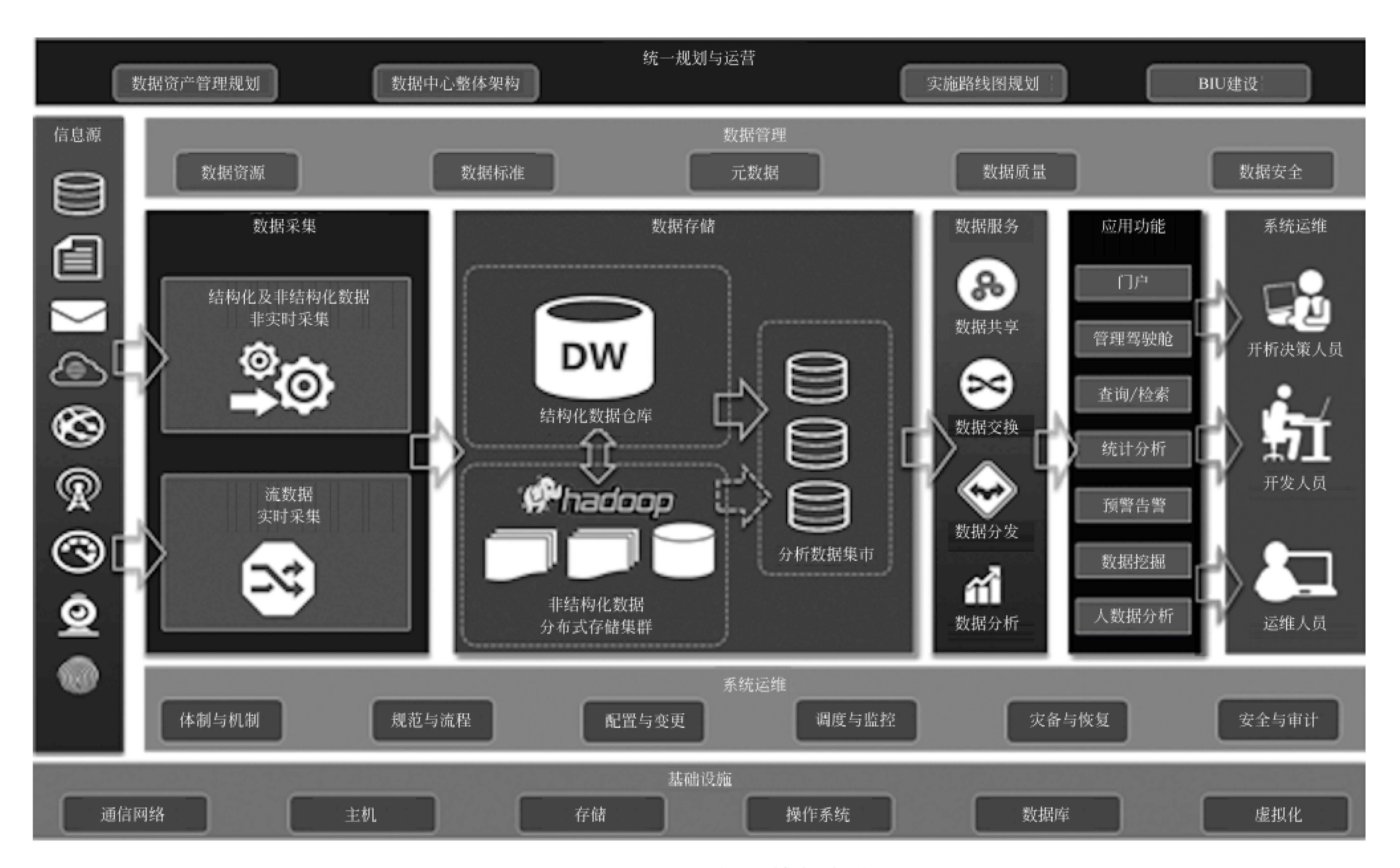


图 11-12　人大金仓大数据架构图

11.6　大数据时代的数据保护

按照通用的大数据架构的理解，我们自底向上，分别讨论一下文件系统、数据库、分布式服务框架、私有云等各个层面的数据保护。

11.6.1　HDFS

Hadoop 分布式文件系统（HDFS）被设计成适合运行在通用硬件上的分布式文件系统。HDFS 是一个高度容错性的系统，适合部署在廉价的机器上。HDFS 能提供高吞吐量的数

据访问,非常适合大规模数据集上的应用。HDFS放宽了一部分POSIX约束,来实现流式读取文件系统数据的目的。

HDFS有着高容错性的特点,并且设计用来部署在低廉的硬件上。而且它提供高吞吐量(high throughput)来访问应用程序的数据,适合那些有着超大数据集(large data set)的应用程序。

1. 数据复制

HDFS设计成能可靠地在集群中大量机器之间存储大量的文件,它以块序列的形式存储文件。文件中除了最后一个块,其他块都有相同的大小。文件中的块为了故障容错而被复制。块的大小和复制数是以文件为单位进行配置的,应用可以在文件创建时或者之后修改复制因子。HDFS中的文件是一次写的,并且任何时候都只有一个写操作。

名字节点负责处理所有的块复制相关的决策。它周期性地接受集群中数据节点的心跳和块报告。一个心跳的到达表示这个数据节点是正常的。一个块报告包括该数据节点上所有块的列表。

默认的HDFS block放置策略在最小化写开销和最大化数据可靠性、可用性以及总体读取带宽之间进行了一些折中。一般情况下复制因子为3,HDFS的副本放置策略是将第一个副本放在本地节点,将第二个副本放到本地机架上的另外一个节点,而将第三个副本放到不同机架上的节点。这种方式减少了机架间的写流量,从而提高了写的性能。机架故障的概率远小于节点故障。这种方式并不影响数据可靠性和可用性的限制,并且它确实减少了读操作的网络聚合带宽,因为文件块仅存在两个不同的机架,而不是三个。文件的副本不是均匀地分布在机架当中,1/3在同一个节点上,1/3副本在同一个机架上,另外1/3均匀地分布在其他机架上。这种方式提高了写的性能,并且不影响数据的可靠性和读性能。

为了尽量减小全局的带宽消耗读延迟,HDFS尝试返回给一个读操作离它最近的副本。假如在读节点的同一个机架上就有这个副本,就直接读这个,如果HDFS集群是跨越多个数据中心,那么本地数据中心的副本优先于远程的副本。

一个数据节点周期性发送一个心跳包到名字节点。网络断开会造成一组数据节点子集和名字节点失去联系。名字节点根据缺失的心跳信息判断故障情况。名字节点将这些数据节点标记为死亡状态,不再将新的I/O请求转发到这些数据节点上,这些数据节点上的数据将对HDFS不再可用,可能会导致一些块的复制因子降低到指定的值。

名字节点检查所有的需要复制的块,并开始复制它们到其他的数据节点上。重新复制在有些情况下是不可或缺的,例如:数据节点失效,副本损坏,数据节点磁盘损坏或者文件的复制因子增大。

HDFS NameNode的高可用整体架构如图11-13所示。

可以看出NameNode的高可用架构主要分为下面几个部分。

(1) Active NameNode和Standby NameNode

两台NameNode形成互备,一台处于Active状态,为主NameNode;另外一台处于Standby状态,为备用NameNode。只有主NameNode才能对外提供读写服务。

(2) 主备切换控制器ZKFailoverController

ZKFailoverController作为独立的进程运行,对NameNode的主备切换进行总体控制。ZKFailoverController能及时检测到NameNode的健康状况,在主NameNode故障时借助

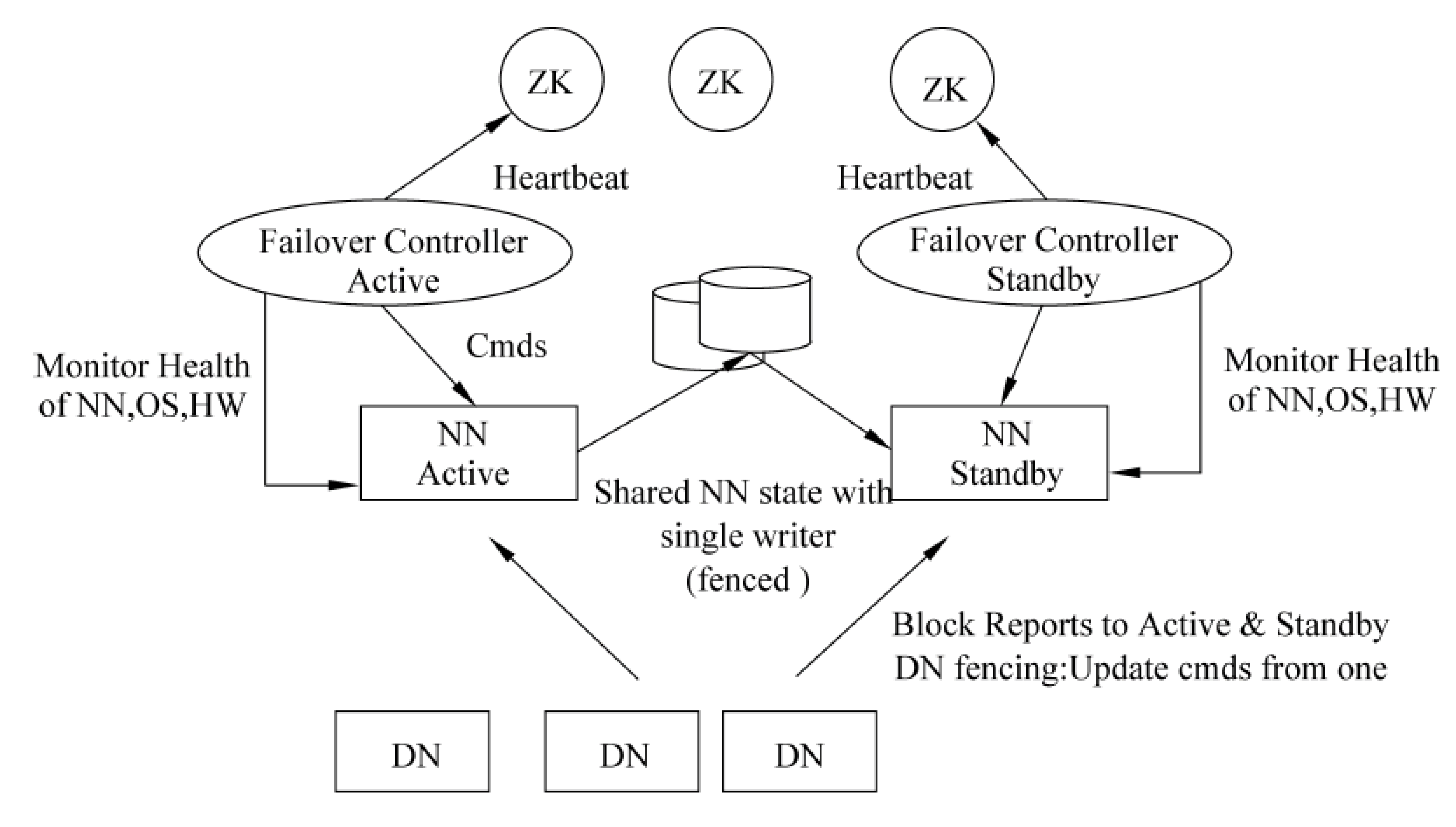


图 11-13　HDFS NameNode 的高可用整体架构

Zookeeper 实现自动的主备选举和切换，当然 NameNode 目前也支持不依赖于 Zookeeper 的手动主备切换。

(3) Zookeeper 集群

为主备切换控制器提供主备选举支持。

(4) 共享存储系统

共享存储系统是实现 NameNode 的高可用最为关键的部分，共享存储系统保存了 NameNode 在运行过程中所产生的 HDFS 的元数据。主 NameNode 和备用 NameNode 通过共享存储系统实现元数据的同步。在进行主备切换的时候，新的主 NameNode 在确认元数据完全同步之后才能继续对外提供服务。

DataNode 节点除了通过共享存储系统共享 HDFS 的元数据信息之外，主 NameNode 和备用 NameNode 还需要共享 HDFS 的数据块和 DataNode 之间的映射关系。DataNode 会同时向主 NameNode 和备用 NameNode 上报数据块的位置信息。

2. 验证数据的正确性

从数据节点上取一个文件块有可能是坏块，坏块的出现可能是存储设备错误、网络错误或者软件的漏洞。HDFS 客户端实现了 HDFS 文件内容的校验。当一个客户端创建一个 HDFS 文件时，它会为每一个文件块计算一个校验码并将校验码存储在同一个 HDFS 命名空间下一个单独的隐藏文件中。当客户端访问这个文件时，它根据对应的校验文件来验证从数据节点接收到的数据。如果校验失败，客户端可以选择从其他拥有该块副本的数据节点获取这个块。

11.6.2　HBase

HBase 是一个没有单点故障的分布式系统，上层(HBase 层)和底层(HDFS 层)都通过一定的技术手段保障了服务的可用性。上层 HMaster 一般都是高可用部署，而 RegionServer 如果出现宕机，region 迁移的代价并不大，一般都在毫秒级别完成，所以对应用造成的影响也很有限；底层存储依赖于 HDFS，数据本身默认也有三个副本，数据存储上

做到了多副本冗余，而且 Hadoop 2.0 以后 NameNode 的单点故障也被消除。其原理如下。

HBase 高可用保证在出现异常时，快速进行故障转移。下面让我们先来看看 HBase 高可用的实现，如图 11-14 所示。

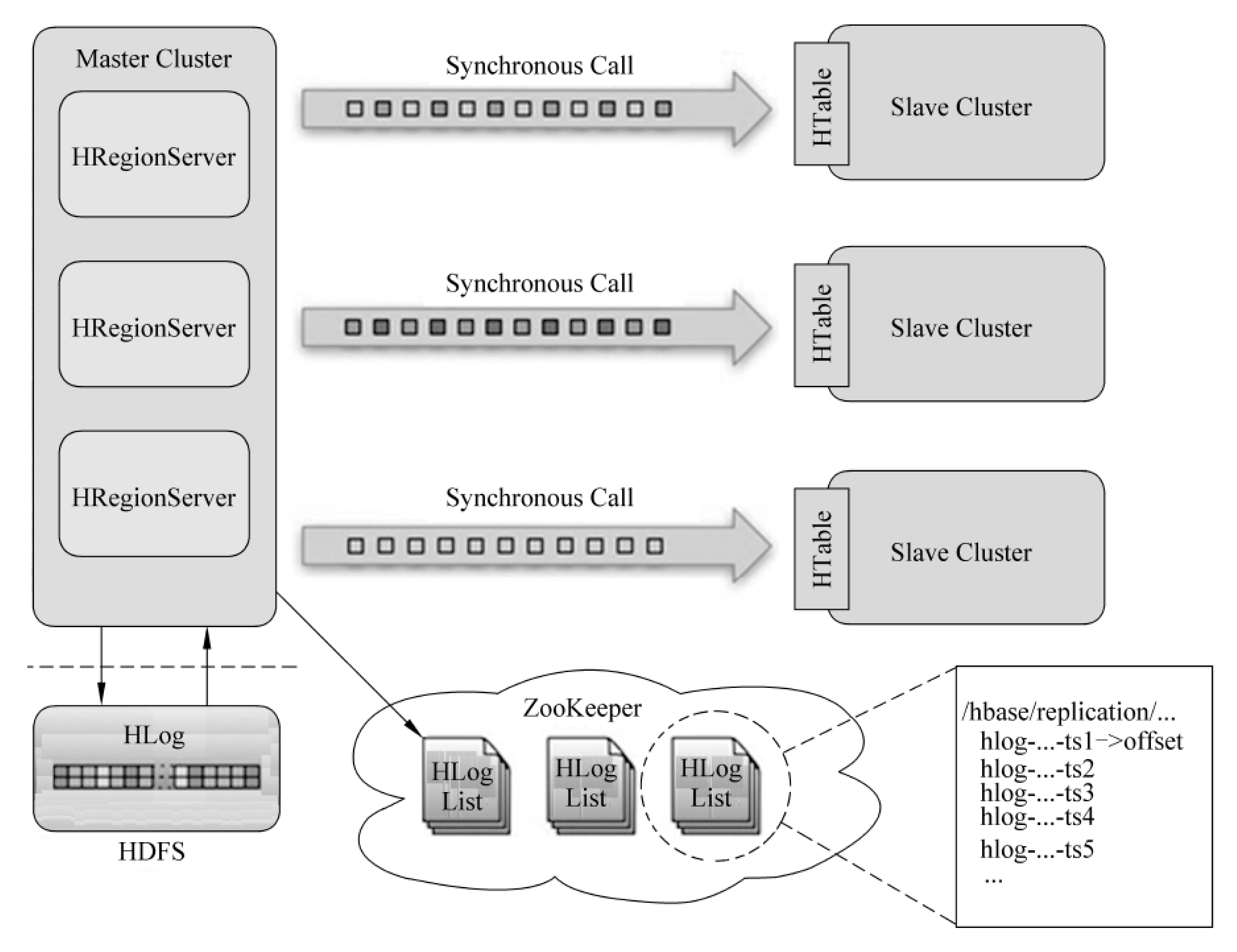


图 11-14　HBase 高可用示意图

图中一个 Master 对应了 3 个 Slave，Master 上每个 RegionServer 都有一份 HLog，在开启 Replication 的情况下，每个 RegionServer 都会开启一个线程用于读取该 RegionServer 上的 HLog，并且发送到各个 Slave，Zookeeper 用于保存当前已经发送的 HLog 的位置。Master 与 Slave 之间采用异步通信的方式，保障 Master 上的性能不会受到 Slave 的影响。用 Zookeeper 保存已经发送 HLog 的位置，主要考虑在 Slave 复制过程中如果出现问题后重新建立复制，可以找到上次复制的位置。

HBase 同步复制步骤如下。

① HBase Client 向 Master 写入数据。

② RegionServer 写完 HLog 后返回 Client 的请求。

③ replication 线程轮询 HLog 发现有新的数据，发送给 Slave。

④ Slave 处理完数据后返回给 Master。

⑤ Master 收到 Slave 的返回信息，在 Zookeeper 中标记已经发送到 Slave 的 HLog 位置。

如图 11-15 演示了上述步骤。

HBase 通过 HLog 进行数据复制，从复制模式上来讲，HBase 支持主从、主主两种复制模式，也就是经常说的 Master-Slave、Master-Master 复制。

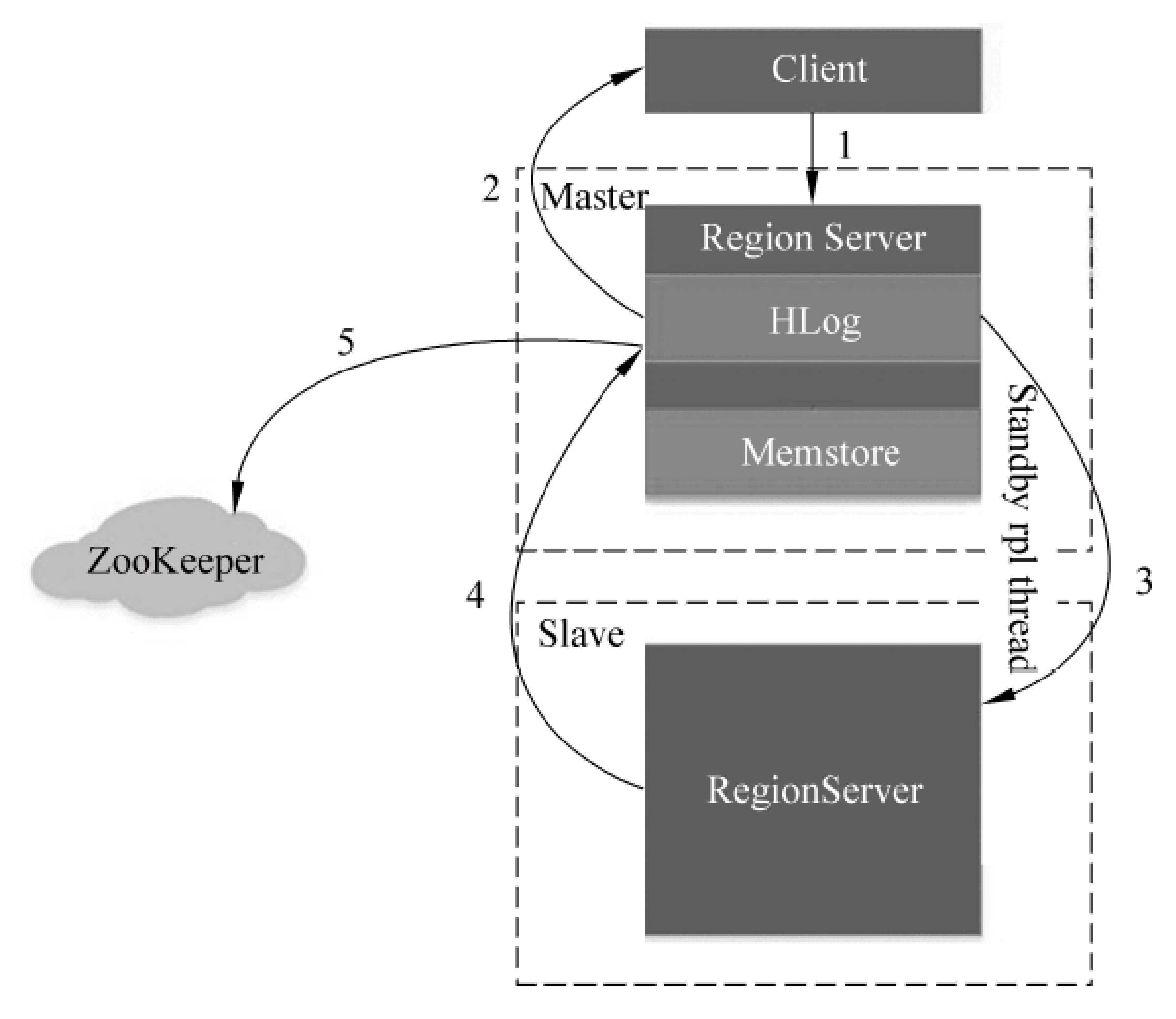


图 11-15　HBase 同步复制步骤示意图

11.6.3　Zookeeper

Zookeeper 是针对大型分布式系统的高可靠的协调系统，它主要是用来解决分布式应用中经常遇到的一些数据管理问题，如：统一命名服务、状态同步服务、集群管理、分布式应用配置项的管理等。

开发分布式系统是件很困难的事情，其中的困难主要体现在分布式系统的“部分失败”。“部分失败”是指信息在网络的两个节点之间传送时，如果网络出了故障，发送者无法知道接收者是否收到了这个信息，而且这种故障的原因很复杂，接收者可能在出现网络错误之前已经收到了信息，也可能没有收到，或者接收者的进程死掉了。发送者能够获得真实情况的唯一办法就是重新连接到接收者，询问接收者错误的原因，这就是分布式系统开发里的“部分失败”问题。

Zookeeper 就是解决分布式系统“部分失败”的框架。Zookeeper 不是让分布式系统避免“部分失败”问题，而是让分布式系统碰到部分失败时，可以正确地处理此类的问题，让分布式系统能正常地运行。

1. 统一命名服务（Name Service）

分布式应用中，通常需要有一套完整的命名规则，既能够产生唯一的名称又便于人识别和记住，通常情况下用树形的名称结构是一个理想的选择，树形的名称结构是一个有层次的目录结构，既对人友好又不会重复。Zookeeper 的 Name Service 与 JNDI 能够完成的功能是差不多的，它们都是将有层次的目录结构关联到一定资源上，但是 Zookeeper 的 Name Service 更加是广泛意义上的关联，也许并不需要将名称关联到特定资源上，可能只需要一个不会重复名称，就像数据库中产生一个唯一的数字主键一样。

Name Service 已经是 Zookeeper 内置的功能，只要调用 Zookeeper 的 API 就能实现。如调用 create 接口就可以很容易创建一个目录节点。

2. 配置管理(Configuration Management)

配置的管理在分布式应用环境中很常见,例如同一个应用系统需要多台 PC Server 运行,但是它们运行的应用系统的某些配置项是相同的,如果要修改这些相同的配置项,那么就必须同时修改每台运行这个应用系统的 PC Server,这样非常麻烦而且容易出错。

类似这样的配置信息完全可以交给 Zookeeper 来管理,将配置信息保存在 Zookeeper 的某个目录节点中,然后将所有需要修改的应用机器监控配置信息的状态,一旦配置信息发生变化,每台应用机器就会收到 Zookeeper 的通知,然后从 Zookeeper 获取新的配置信息应用到系统中,如图 11-16 所示。

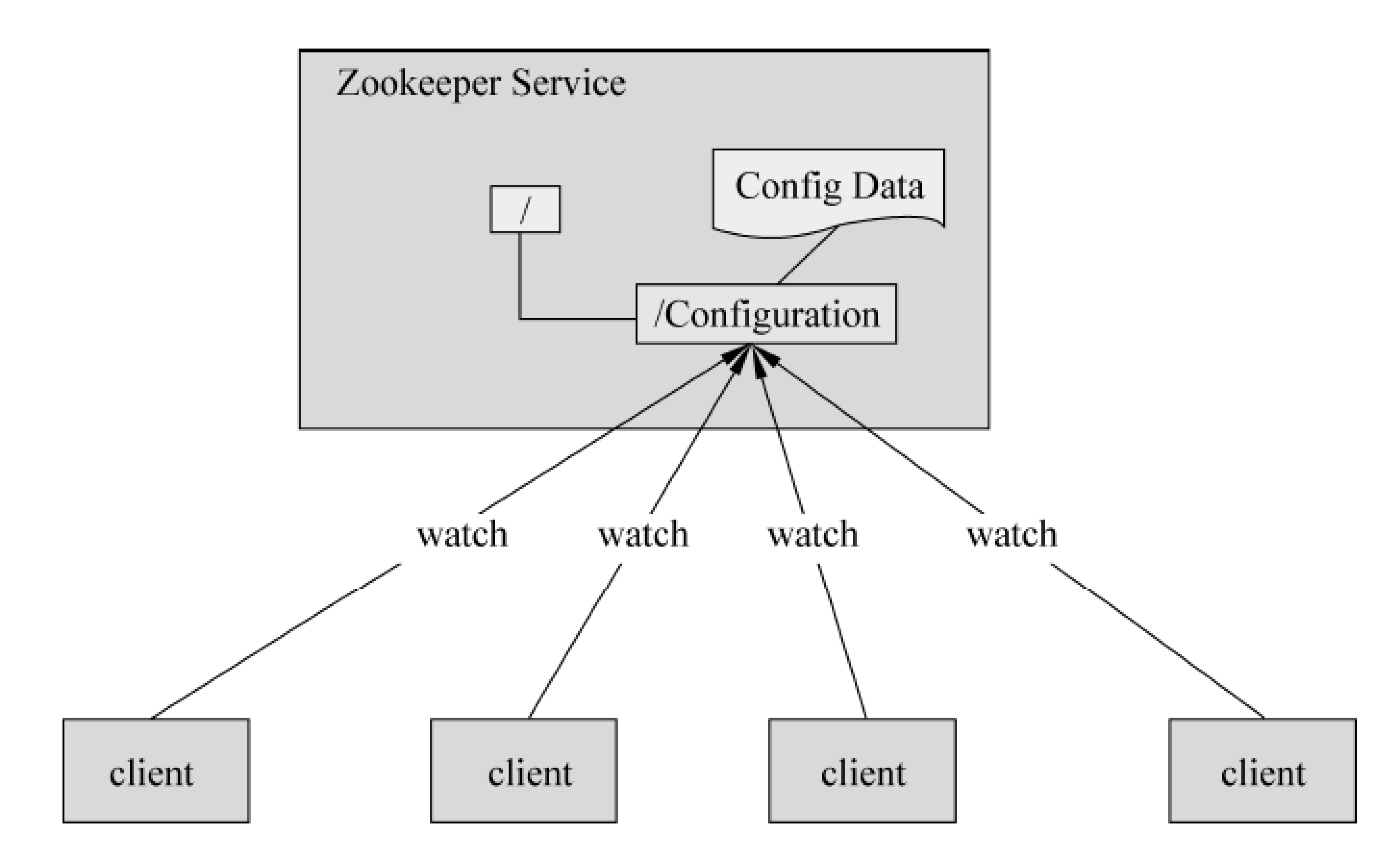


图 11-16　Zookeeper 配置管理结构图

3. 集群管理(Group Membership)

Zookeeper 能够很容易地实现集群管理的功能,如有多台 Server 组成一个服务集群,那么必须要一个"总管"知道当前集群中每台机器的服务状态,一旦有机器不能提供服务,集群中其他集群必须知道,从而做出调整并重新分配服务策略。同样当增加集群的服务能力时,就会增加一台或多台 Server,同样也必须让"总管"知道。

Zookeeper 不仅能够维护当前的集群中机器的服务状态,而且能够选出一个"总管",让这个总管来管理集群,这就是 Zookeeper 的另一个功能 Leader Election。

4. 共享锁(Locks)

共享锁在同一个进程中很容易实现,但是在跨进程或者在不同 Server 之间就不好实现了。Zookeeper 却很容易实现这个功能。实现方式是需要获得锁的 Server 创建一个 EPHEMERAL_SEQUENTIAL 目录节点,然后调用 getChildren 方法获取当前的目录节点列表中最小的目录节点是不是就是自己创建的目录节点,如果正是自己创建的,那么它就获得了这个锁;如果不是,它就调用 exists(String path, boolean watch)方法并监控 Zookeeper 上目录节点列表的变化,一直到自己创建的节点是列表中最小编号的目录节点,从而获得锁。释放锁很简单,只要删除前面它自己所创建的目录节点就行了。

11.6.4　OpenStack

1. OpenStack 架构与 HA 分析

OpenStack 实际上是由众多服务组合而成,它们之间或多或少有关联,而且具有一定的

层次关系，每个服务就像积木块一样，你可以根据实际需要进行取舍并组合搭建，因此良好的运营架构整合能力是应用 OpenStack 的前提。

在 OpenStack 的计算、网络和存储服务分别对应的是 Nova、Neutron、Cinder 这几个服务。从社区给出的 OpenStack 各个服务的应用统计来看，也是这几个服务接受程度最高，也相对最成熟，另外，从目前 OpenStack 生态去看，Swift 的接受程度并不高，一个重要原因是 Ceph 在云计算领域的开疆拓土一定程度上挤占了 Swift 的市场。相比 Swift 而言，Ceph 是一个大一统的存储解决方案，在对象存储、块存储、文件存储三大方向都能够由 Ceph 底层的 Rados 实现，虽然 Ceph Rados 不具备数据排重等高级功能，在落地存储上也没有自己很核心的技术，但是在整个架构的 Scaling 和 HA 处理方面做得相当不错，其设计理念比代码实现要超前。统一起来相当方便，而这三者恰恰是任何一个通用云计算平台所需要的。

对任何一个分布式系统，高可用 HA 都是最核心的设计目标之一。而 OpenStack 这样一个复杂系统，高可用更涉及多个层面，只要有一个层面做不到，那么整个 OpenStack 都没法实现高可用。如图 11-17 所示。

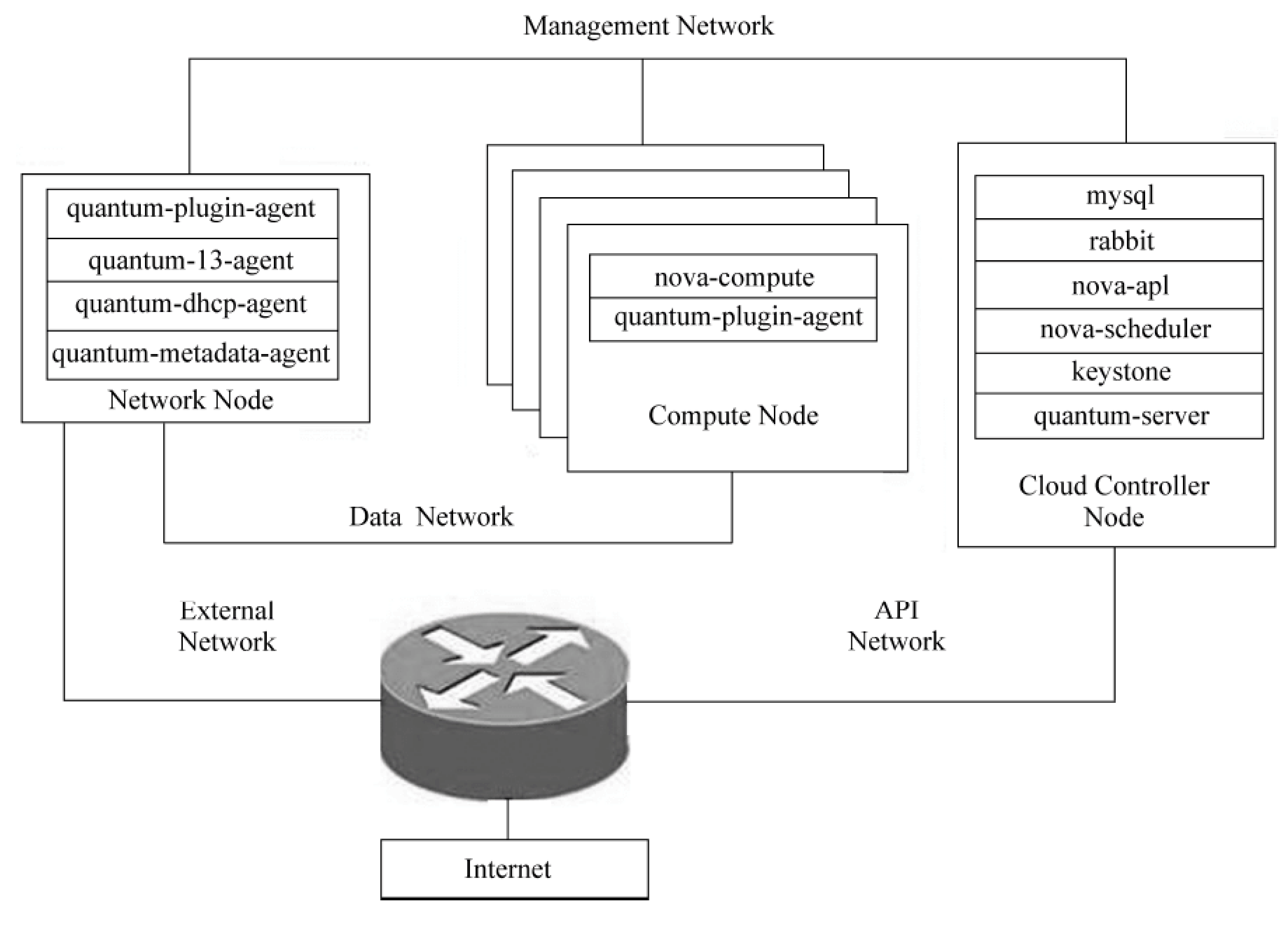


图 11-17　OpenStack 物理架构示意图

所以 OpenStack 的高可用可以从两个层面去划分，从功能服务层面划分为：

- 基础服务（mysql，rabbitmq）；
- 计算（nova）；
- 网络（neutron）；
- 存储（cinder）。

从物理部署层面划分为：

- 控制节点（主要部署基础服务＋其他服务的接入、调度模块）；

- 网络节点(主要部署 Neutron 的 L2/L3/DHCP Agent,DHCP,Virtual Router);
- 计算节点(Nova ComputeAgent,Neutron L2Agent,虚拟机)。

不管从那个层面去划分,都需要确保在每个层面上的高可用,并且在各个层面之间进行有效衔接。

在 HA 设计中,一般来说无状态的模块处理是比较简单的,基本思路是并行运行多个节点或者服务模块且对它们进行负载均衡。典型例子是一个网站的 Web 服务器集群,往往采用前端加 LVS 或者 Nginx 之类的 LoadBanlace 服务器解决 HA 问题。LVS 和 Nginx 的高可用主要是利用 Keepalived、Heartbeat 等基于路由冗余协议 VRRP 或心跳仲裁机制来解决。

而对于有状态的模块,主要有两种方式来实现 HA,一种是多节点基于分布式一致性协议(比如 Paxos、Raft 协议等)维护相同的状态,典型的代表有 Zookeeper、Rabbitmq;另一种是基于主从模式的同步或异步复制来维护相同的状态,比如 Mysql、Redis。这两种方式前者较复杂,在一些场景下性能会很低,后者在数据一致性和伸缩性方面有所不足。

如前面提到 OpenStack 的情况会比较复杂,实际实践中这两种都会混合使用,另外有两点我们可以姑且不考虑。

(1) 计算节点,主要涉及虚拟机的可用性,而虚拟机的可用性实际上是跟上层应用密切相关的(要做到一个虚拟机严格的热备是很困难的,存储容易做到,但是 CPU 和内存就难了,所以主要还是靠上层应用处理),而且对于上层应用来说可能并不需要,应用可能有助于业务逻辑的容错设计。

(2) 存储方面,Cinder 虽然是 OpenStack 的存储服务,但是跟 Swift 不同,打个比方,Cinder 只是一个存储管理器而不是存数据的“硬盘”,真正的“硬盘”是底层的 LVM、Ceph、GlusterFS 以及其他软件或硬件构成的存储系统等,所以 OpenStack 在存储方面的高可用更多的是指 Cinder 这个管理器的高可用性,而数据存储的高可用性已经由底层的存储系统来解决了(比如 Ceph)。

综合上述分析,OpenStack 的高可用,主要是确保控制节点和网络节点的高可用,映射到功能服务维度上,就是确保基础服务(Mysql 和 Rabbitmq)高可用,Nova、Neutron 和 Cinder 的接入与调度高可用,以及 Neutron 所创建的 DHCP 和 Virtual Router 等虚拟网络设施的高可用。下面逐一进行探讨。

2. OpenStack 各层次的 HA 设计

(1) 基础服务 Mysql 和 RabbitMQ

MySQl 作为开源 DBMS 已经是相当成熟了,功能也非常全面,支持多种数据库表引擎,生态完善,但是如果从分布式数据库系统的角度去看,其实还不是很成熟。目前大家用得最多还是基于 binlog 复制的 Master-Slave 模式进行数据复制,并基于此做高可用和读写分离等设计。比较好用的方案有 MHA。在一主多备的情况下,能够在最少的数据丢失的基础上实现一定的分布式容错与计算,如图 11-18 所示。

不同于 MHA 这种上层的 HA 方案(主要是受限于 MySQL 基于 binlog 的 replication 机制的局限性,在性能和可靠性方面有冲突),在 MySQL 的 MariaDB 和 Percona 分支上,使用兼容 innodb 的 XtraDB 引擎,基于 Galera 集群方式的分布式方案也是越来越受到追捧。虽然复杂度更高,但是分布式实时数据一致性的优势还是非常吸引人的。当然,这种方案有

一些功能上的局限性，另外在写少读多的情况下其实相对 1-Master-N-Slave 架构没有多少优势，如图 11-19 所示。

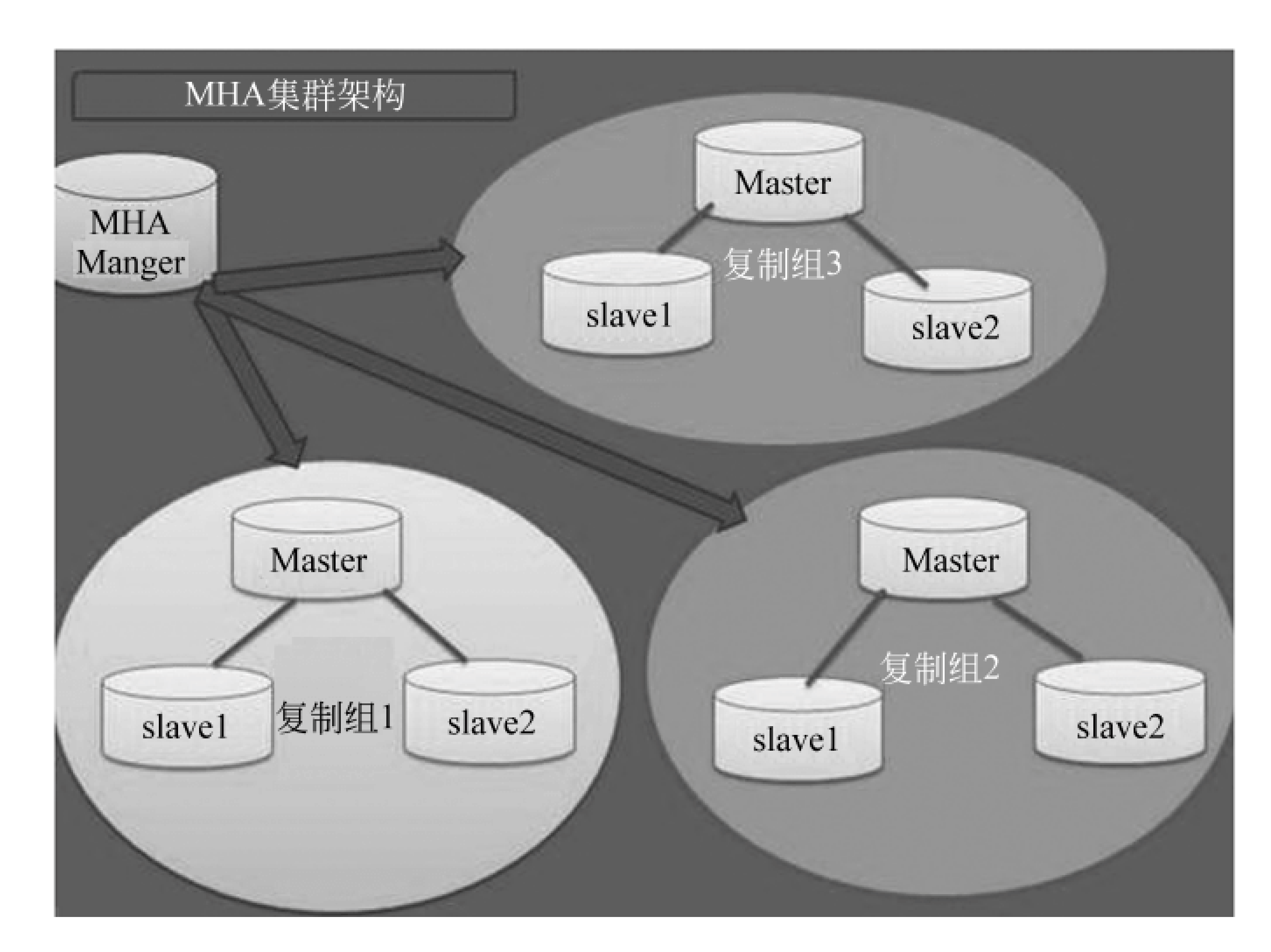


图 11-18　MHA 的典型架构

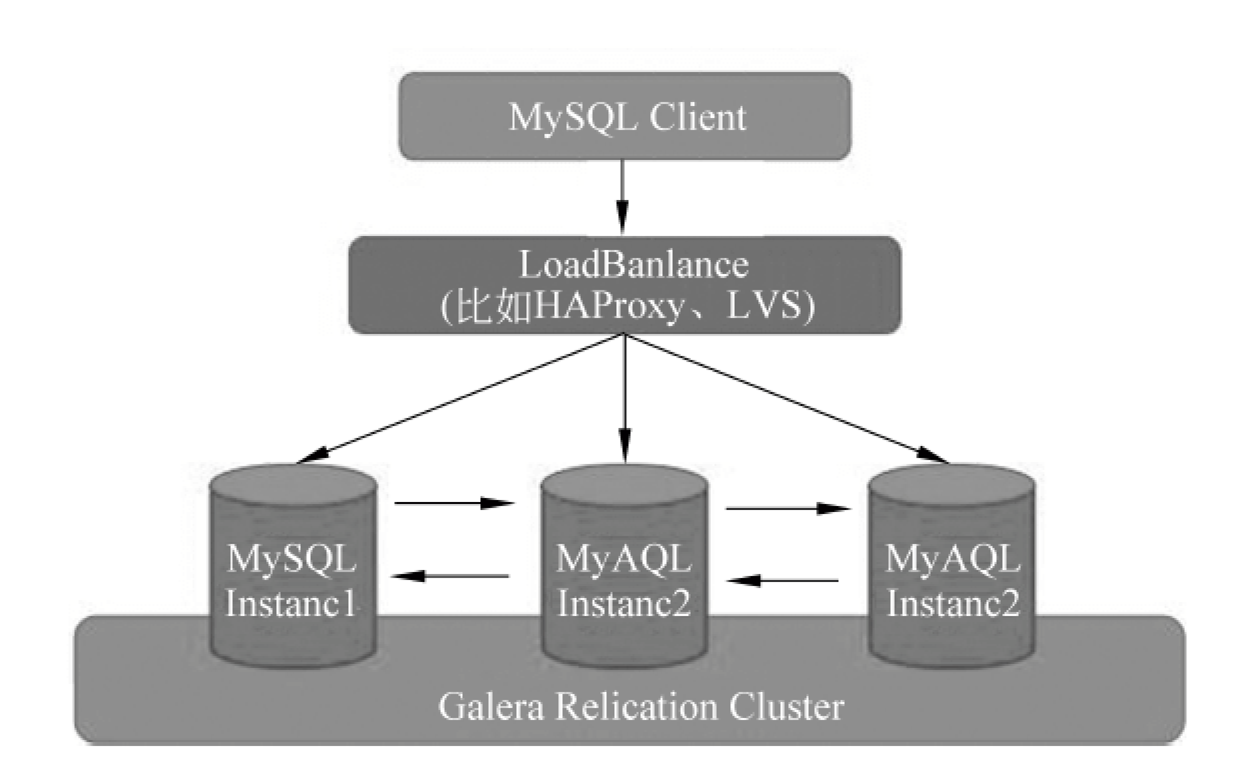


图 11-19　Galera 集群示意图

在开源的分布式消息队列里面，Rabbitmq 算是以稳定可靠而著称，虽然在吞吐量上与 Kafka 族系的消息队列有一些差距，但是经过调优后还是在同一个数量级。

Rabbitmq 内置有 Cluster 集群功能，同一个 Cluster 的节点会共享 topic、exchange、binding 和 queue 等元信息，但是对于真正的 queue 消息数据是要依赖于 Mirror Queue 机制来实现消息的 HA 的，而且组成 Cluster 建议至少要有 3 个节点，否则网络分区发生的时候也不好做决策。

（2）Nova、Neutron、Cinder 接入与控制服务

解决了基础服务后，对于 OpenStack 核心的 Keystone、Nova-API、Nova-Conductor、Nova-Scheduler、Neutron-Server、Cinder-API、Cinder-Scheduler 等，其实都是无状态的，只要多出两个，并且能够做到负载均衡，那么也就基本达成了 HA 的目标了（这里要注意 Nova 的调度和 Cinder 的调度需要进行同步互斥）。考虑到 OpenStack 的对外 API 基本是

HTTP-RESTful 的，所以常见的是采用 Nginx（或 HAProxy）+keepalived（或 PaceMaker）来实现这一层次的 HA 接入，如图 11-20 所示。

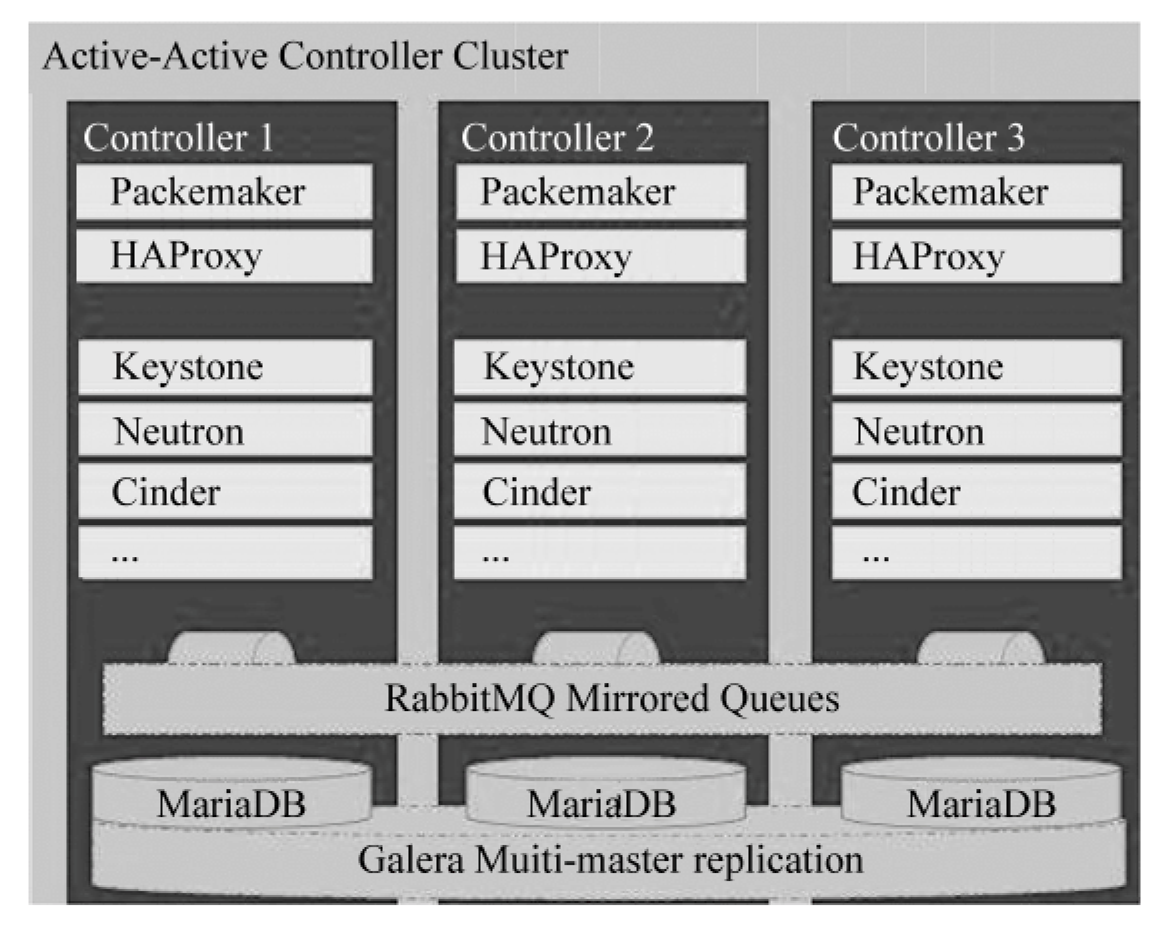


图 11-20　OpenStack 实现 HA 示意图

（3）网络服务

在 OpenStack 中，网络处理占据了相当大的一块，而且由于网络的特殊性与复杂性，一般要独立部署网络节点。网络节点上最核心的就是 L3Agent、DHCPAgent 以及由它们所管理的 DHCP Server 和 Virtual Router 服务。

首先看 DHCP，可以在多个网络节点上部署 DHCP Agent 来达到多 DHCP Server 并行，且把用户私有网络的 DHCP 分布在上面就可以了。

对于 Router 服务，由于涉及路由和外网接入，所以这里不能同时运行多个一样的 Router 服务（地址与路由冲突问题），目前简单的是采取 A/P 模式来部署。由控制节点上的 L3 Router Plugin 去对网络节点上的 L3 Agent 周期性做心跳探测，从而实现 L3 Agent 的 failover 机制，当出现故障时迁移 Router 到新的网络节点上。

从 OpenStack Juno 版本开始引入了分布式虚拟路由 DVR，核心思想是把原来网络节点上的 Router 服务分布到各个计算节点上去了，只把 DHCP 和 SNAT 留在网络节点上。这样就大大增强了 Router 的容灾能力，而且大大增强了整个集群的东西、南北向通信能力（突破了网络节点的瓶颈）。

总而言之，OpenStack 在整体架构上是可以整合出一套行之有效的 HA 方案的。以 OpenStack 为基础，已经整合构建了具有较高可用性的弹性计算、分布式块存储和虚拟私有网络等 IaaS 核心功能。

任务拓展

1. 为什么需要大数据存储技术？
2. 分布式文件系统的意义是什么？
3. 简单介绍大数据存储的关键技术。

参 考 文 献

[1] G. Somasundaram ,Alok Shrivastava. 信息存储与管理：数字信息的存储、管理和保护[M]. 2 版. 马衡，赵甲，译. 北京：人民邮电出版社，2013.
[2] 吴晨涛. 信息存储与 IT 管理[M]. 北京：人民邮电出版社，2015.
[3] 鲍布，鲁迪斯. 数据驱动安全：数据安全分析、可视化和仪表盘[M]. 北京：机械工业出版社，2015.
[4] 盖国强. 数据安全警示录[M]. 北京：电子工业出版社，2012.
[5] 冯登国，张敏，李昊. 大数据安全与隐私保护[J]. 计算机学报，2014(1).
[6] 陈钊. 基于云灾备的数据安全存储关键技术研究[J]. 北京邮电大学学报，2012(2).
[7] 唐芸. 云存储在数字资源长期保存中的应用研究[J]. 湘潭大学学报，2012(3).
[8] 韩丽颖. 基于 Hadoop 技术的轨道交通 MSS 系统数据存储应用研究[J]. 北京交通大学学报，2015(4).
[9] 赵艳玲. 数据存储备份策略及调度研究[J]. 大庆石油学院学报，2008(9).
[10] 金鑫. 数据灾难恢复软件系统的设计与实现[J]. 上海交通大学学报，2012(7).